CLOUDFLOW: PHYSICAL EVOLUTION-GUIDED FLOW MATCHING FOR MULTISPECTRAL GEOSTATIONARY METEOROLOGICAL SATELLITE DATA NOWCASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The increasing frequency of extreme weather events necessitates precise real-time nowcasting for effective disaster prevention and mitigation, a task that relies heavily on high spatiotemporal resolution observations from geostationary meteorological satellites (GMS). However, existing deep learning models face a persistent dilemma where many deterministic methods suffer from severe blurring, while many generative approaches are plagued by severe structural inconsistency and radiance intensity deviations. To overcome this dilemma, we propose CloudFlow, a two-stage framework that constructs a physics-inspired evolution prior and then performs physics-anchored conditional-flow refinement. In the first stage, the proposed Evolution Network is motivated by the continuity equation and implemented with a differentiable semi-Lagrangian transport operator, together with motion and source-sink parameterization. In the second stage, the proposed Physics-Anchored Flow Matching utilizes the coarse-grained prediction of the Evolution Network as a structural anchor to guide the reconstruction of fine-grained details. Extensive experiments on the Himawari-8/9 and GOES-16 datasets show that CloudFlow unifies predictive accuracy, visual fidelity, inference efficiency, robustness and evolution interpretability.

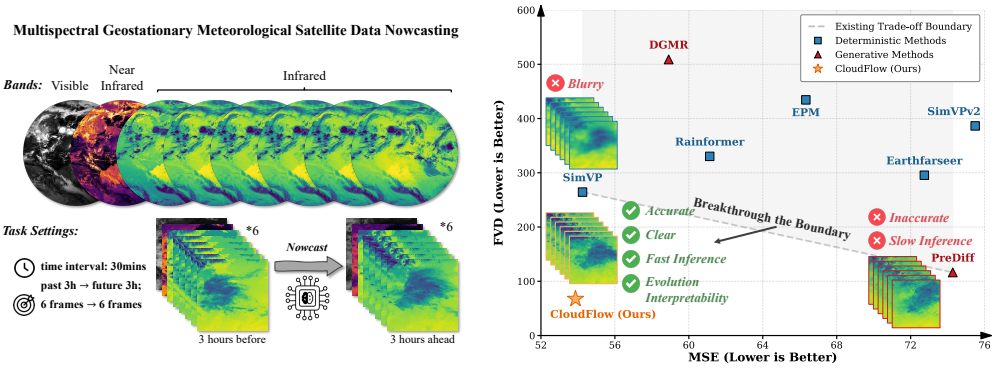


Figure 1: Task illustration (Left) and performance analysis (Right). The task is to predict future sequences from historical multispectral observations, including visible, near-infrared, and infrared bands. The proposed CloudFlow breaks the existing perception-accuracy trade-off boundary, achieving best clarity and accuracy compared to other methods, while maintaining fast inference and ensuring evolution interpretability.

1 INTRODUCTION

The increasing frequency of extreme weather events poses a severe challenge to disaster prevention and mitigation Diffenbaugh et al. (2017), particularly given the sudden onset and rapid evolution of these phenomena Merk & Zinner (2013). To effectively capture these dynamics, Geostationary Meteorological Satellites (GMS), such as the Himawari Bessho et al. (2016) and GOES series Schmit et al.

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.

(2017), serve as the critical observational infrastructure. Distinguished by their high spatiotemporal resolution, these satellites provide continuous monitoring that is essential for tracking multi-scale atmospheric systems Zhang et al. (2022). As shown at the left of Figure 1, unlike standard natural images, GMS observations represent a complex radiance field, measured as reflectance in visible bands and brightness temperature in infrared bands, which offers a direct physical representation of atmospheric thermodynamics Li et al. (2013), including the semantics of cloud morphology and water vapor distribution Bony et al. (2020). Consequently, leveraging this physically informative, high spatiotemporal resolution GMS data to nowcast its evolution over the next 3 hours plays an important role in extreme weather disaster prevention and mitigation.

The advent of deep learning has revolutionized GMS data nowcasting by introducing data-driven spatiotemporal learning, aiming to capture the complex nonlinear dynamics of atmospheric evolution through high-capacity neural architectures. Initial research primarily focused on deterministic models, which treat nowcasting as a regression task. RNN-based architectures, such as ConvLSTM Shi et al. (2015), PredRNN Wang et al. (2022), and EPM Xia et al. (2024), coupled spatial convolutions with temporal recursions to capture local dynamics, while frameworks like PhyDNet Guen & Thome (2020) sought to explicitly disentangle underlying physical components. This trajectory has further expanded with high-capacity convolutional baselines like SimVP Gao et al. (2022a), SimVPv2 Tan et al. (2025), and Transformer-based variants such as Earthformer Gao et al. (2022b), Earthfarseer Wu et al. (2024) and AlphaPre Lin et al. (2025). Undoubtedly, these deterministic models deliver highly accurate predictions by effectively learning complex spatiotemporal patterns. However, their predictions often become overly smooth and lose perceptual quality, indicating that the models fail to reconstruct the fine-grained meteorological details, such as the sharp edges of intense convective cells, which are essential for reliable extreme weather warnings.

To address this pervasive blurriness and to capture the inherent stochasticity of atmospheric evolution, the generative paradigm emerges as a compelling solution. Following its emergence, this approach has evolved from early adversarial models like DGMR Ravuri et al. (2021) to the current prevalence of diffusion-based frameworks like MCVD Voleti et al. (2022) and PreDiff Gao et al. (2023). By approximating the score function of the data distribution, these diffusion-based models transform Gaussian noise, conditioned on historical observations, into potential future sequences through an iterative denoising process, thereby producing clear predictions with high perceptual quality. However, this paradigm introduces new critical challenges. The unconstrained stochasticity of the sampling process frequently leads to structural inconsistency and deviations in the radiance intensity Daras et al. (2023). Furthermore, the multi-step iterative nature of diffusion-based inference incurs substantial computational cost, presenting a practical challenge for operational efficiency.

To break this perception–accuracy trade-off, we propose CloudFlow, a two-stage framework for multispectral GMS nowcasting. CloudFlow predicts 8-band GMS sequences for six future steps at 30-minute intervals. It first models physical evolution to obtain a deterministic coarse prediction and then performs physics-anchored conditional-flow refinement to recover fine-grained details. Although the refinement stage adopts a flow-matching training objective, its inference process involves no noise sampling and produces a unique and deterministic prediction for a given input. This design preserves accuracy while improving visual fidelity, structural stability, and computational efficiency. The main contributions are:

- **CloudFlow** is proposed as a novel two-stage framework that integrates deterministic physical evolution with physics-anchored conditional-flow refinement, which effectively breaks through the persistent perception–accuracy trade-off in multispectral GMS data nowcasting.
- **Evolution Network** inspired by physical advection is proposed to explicitly decompose cloud evolution into motion-driven transport and source-sink intensity variation, thereby providing interpretable evolution priors.
- **Physics-Anchored Flow Matching** mechanism is designed to utilize coarse-grained physical predictions as structural anchors, significantly enhancing the precision of fine-grained detail reconstruction while reducing computational cost.
- Extensive experiments demonstrate that CloudFlow outperforms existing state-of-the-art methods in **accuracy and visual sharpness**, while maintaining **robustness** across diverse scenarios, particularly in capturing localized extreme weather such as **rapid convective developments**.

2 TASK DEFINITION

Following the standard formulation of spatiotemporal sequence prediction, we define the GMS data nowcasting task as predicting future sequences based on historical observations. Given a sequence of L past satellite frames $\mathcal{F}_{past} = \{\mathbf{X}_{t-L+1}, \ldots, \mathbf{X}_{t-1}, \mathbf{X}_t\}$, where each frame $\mathbf{X}_i \in \mathbb{R}^{C\times H\times W}$ represents the observed satellite data with C spectral channels at a resolution of $H \times W$, the objective is to predict the subsequent M future frames $\hat{\mathcal{F}}_{pred} = \{\hat{\mathbf{X}}_{t+1}, \hat{\mathbf{X}}_{t+2}, \ldots, \hat{\mathbf{X}}_{t+M}\}$. Formally, we seek to learn an optimal parameterized mapping function f_θ that minimizes the discrepancy between the predicted sequence and the ground-truth future sequence $\mathcal{F}_{GT}$, such that $\hat{\mathcal{F}}_{pred} = f_\theta(\mathcal{F}_{past})$.

3 METHODOLOGY

As shown in Figure 2, CloudFlow is a two-stage framework designed for multispectral GMS data nowcasting. In the first **Physical Evolution** stage, the Evolution Network produces a deterministic intermediate prediction $\hat{\mathcal{F}}_{phy}$ that serves as a coarse-grained prediction for atmospheric dynamics. In the second stage, **Physics-Anchored Flow Matching** (PA-FM) performs deterministic conditional-flow refinement initialized from the deterministic physics-inspired prior $\hat{\mathcal{F}}_{phy}$ rather than sampling from unconditioned noise. The key components are introduced in the subsequent sections. For complete implementation details, please refer to Appendix B.

3.1 PHYSICAL EVOLUTION

Satellite-observed reflectance and brightness temperature are radiometric state variables rather than conserved physical densities. Therefore, the goal is not to solve a strict conservation law like ClimODE Verma et al. (2024), but to introduce a structured evolution prior that separates apparent cloud transport from non-advective radiometric changes. Motivated by the advective form of the continuity equation:

$$\frac{\partial \mathbf{X}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{X}) = \mathbf{S}, \tag{1}$$

we adopt a stable semi-Lagrangian formulation. Backward trajectory tracing yields:

$$\mathbf{X}_{t+1}(\mathbf{p}) = \mathbf{X}_t(\mathbf{p} - \mathbf{v}(\mathbf{p})) + \mathbf{S}(\mathbf{p}). \tag{2}$$

where $\mathbf{X}_t \in \mathbb{R}^{C\times H\times W}$ denotes the multispectral observation at time t, $\mathbf{p}$ is a spatial location, $\mathbf{v}(\mathbf{p})$ represents the predicted velocity field, and $\mathbf{S}(\mathbf{p})$ models the source-sink intensity changes.

Gradient-Enhanced Encoder. We employ a dual-stream Convolutional Neural Network (CNN) encoder to concurrently process historical data $\mathcal{F}_{past}$ and its spatial gradients $\nabla\mathcal{F}_{past}$. Encoding high-gradient cloud boundaries as structural priors allows the model to better resolve discontinuities in the velocity fields v and source-sink term S, particularly within complex convective systems. The encoding process can be formulated as:

$$z_{evo} = \text{Encoder}(\mathcal{F}_{past}, \nabla\mathcal{F}_{past}), \tag{3}$$

where z_{evo} represents the latent feature that serves as the basis for motion and source-sink decoding.

Motion Decoder. Motivated by the common separation between large-scale coherent transport and local deformation in atmospheric motion, we parameterize the apparent velocity field as the sum of a low-dimensional bulk component $\bar{v}$ and a dense residual component v':

$$v = \bar{v} + v'. \tag{4}$$

The **Global Motion Head** is intended to model large-scale atmospheric dynamics. It captures a low-dimensional bulk displacement pattern, including approximate translation and rotation of large-scale cloud systems. For any spatial coordinate (i, j), this transformation is analytically formulated as:

$$\bar{v}(i,j) = \begin{bmatrix} \cos\theta - 1 & -\sin\theta \\ \sin\theta & \cos\theta - 1 \end{bmatrix} \begin{bmatrix} i - c_x \\ j - c_y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}, \tag{5}$$

where θ represents the rotation angle, (c_x, c_y) denotes the rotation center, and (d_x, d_y) signifies the global translation vector. As the latent feature z_{evo} captures high-level abstractions and global

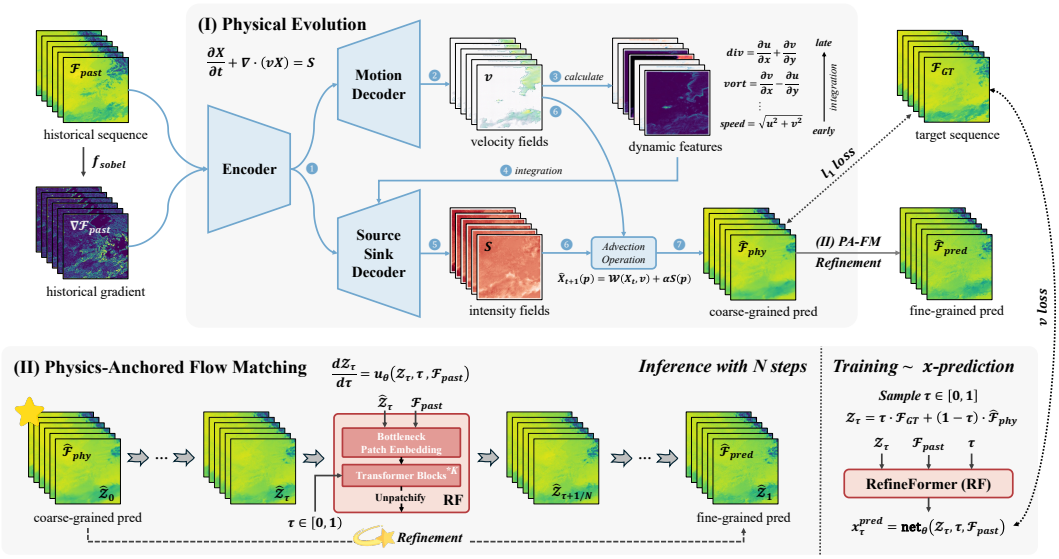


Figure 2: Overview of the proposed CloudFlow framework. The framework consists of two primary stages: (I) Physical Evolution and (II) Physics-Anchored Flow Matching (PA-FM). In the first stage, an encoder extracts spatiotemporal features from historical sequences and gradients to drive a Motion Decoder and a Source-Sink Decoder. These modules explicitly model the velocity fields v and residual fields S inspired by the advective form of atmospheric evolution to produce a physics-informed coarse-grained prediction $\hat{\mathcal{F}}_{phy}$. In the second stage, the PA-FM utilizes $\hat{\mathcal{F}}_{phy}$ as a structural anchor rather than starting from pure noise. A RefineFormer is employed to learn the conditional refinement path and iteratively refine the coarse prediction into the final fine-grained prediction $\hat{\mathcal{F}}_{pred}$.

context, it is naturally suited for extracting spatially-uniform motion parameters. Consequently, the Global Motion Head regresses these parameters directly from z_{evo} through a Global Average Pooling (GAP) layer followed by a fully connected (FC) network:

$$\{\theta, c_x, c_y, d_x, d_y\} = \text{Decoder}_{GM}(z_{evo}) = \text{FC}(\text{GAP}(z_{evo})). \tag{6}$$

The **Local Motion Decoder** is designed to estimate the local residual displacement component v', accounting for complex local advection that deviates from the background flow. Unlike the global parameters, v' is learned by mapping the latent representation z_{evo} back to the spatial domain through a series of CNN-based up-sampling layers and skip connections:

$$v' = \text{Decoder}_{LM}(z_{evo}). \tag{7}$$

Source-Sink Decoder with Dynamics Feature Integration. To characterize the non-conservative evolution of atmospheric fields, the **Source-Sink Decoder** estimates the term S by integrating kinematic descriptors derived from the velocity fields v. We compute a comprehensive set of diagnostic variables representing the fundamental kinematic characteristics of the atmospheric flow:

$$\begin{aligned} z_{\text{vort}} &= \partial_x v_y - \partial_y v_x, & z_{\text{div}} &= \partial_x v_x + \partial_y v_y, \\ z_{e_1} &= \partial_x v_x - \partial_y v_y, & z_{e_2} &= \partial_x v_y + \partial_y v_x, \\ z_{\text{shear}} &= \sqrt{z_{e_1}^2 + z_{e_2}^2}, & z_{\text{strain}} &= \sqrt{z_{\text{div}}^2 + z_{\text{shear}}^2}, \\ z_{\text{speed}} &= \sqrt{v_x^2 + v_y^2}. \end{aligned} \tag{8}$$

where z_{vort} and z_{div} capture cyclonic rotation and convection; z_{e_1}, z_{e_2}, and z_{shear} describe elongation-related shearing; while z_{strain} and z_{speed} quantify total deformation and kinetic intensity Holton & Hakim (2013). To model these hierarchical atmospheric dynamics, we implement a multistage

Dynamics Feature Integration mechanism. Kinematic variables are categorized by their physical roles and integrated at corresponding resolutions:

$$\begin{aligned} \mathbf{E}_{early} &= \{z_{\text{speed}}, z_{e_1}, z_{e_2}, z_{\text{shear}}\}, \\ \mathbf{E}_{mid} &= \{z_{\text{vort}}\}, \\ \mathbf{E}_{late} &= \{z_{\text{div}}, z_{\text{strain}}\}. \end{aligned} \tag{9}$$

We use these kinematic descriptors as scale-aware proxies rather than direct physical measurements. Speed and deformation-related features $\mathbf{E}_{early}$ are injected into deeper layers to provide coarse motion context, while divergence-related features $\mathbf{E}_{late}$ are injected into shallower layers as cues for localized intensity changes associated with cloud growth or dissipation. Vorticity features $\mathbf{E}_{mid}$ are injected at intermediate levels, as they mainly characterize mesoscale rotational structures between coarse advection and local intensity variation. The final source-sink term is decoded as a temporal sequence spanning M frames:

$$S = \text{Decoder}_S(z_{evo}, \mathbf{E}_{early}, \mathbf{E}_{mid}, \mathbf{E}_{late}). \tag{10}$$

Advection Operation. Finally, the predicted velocity v and source-sink term S are integrated through a differentiable semi-Lagrangian warping operation. For each pixel coordinate $\mathbf{p}$ at time t, the predicted field is obtained by:

$$\hat{\mathbf{X}}_{t+1}(\mathbf{p}) = \mathcal{W}(\mathbf{X}_t, v_t) + \alpha S_t(\mathbf{p}), \tag{11}$$

where $\mathcal{W}$ applies bilinear sampling to warp $\mathbf{X}_t$ along v for physical advection, while the scale factor α modulates S to account for radiance change.

3.2 Physics-Anchored Flow Matching

The evolution prior $\hat{\mathcal{F}}_{phy}$ provides a structurally aligned but coarse prediction, which may still miss fine-scale cloud textures and localized intensity details. To refine it, we propose **Physics-Anchored Flow Matching** (PA-FM), which uses $\hat{\mathcal{F}}_{phy}$ instead of Gaussian noise as the starting state and learns a prior-to-data trajectory. This design encourages the vector field to focus on residual correction rather than reconstructing the future sequence from noise, and is implemented with x-prediction parameterization Li & He (2025) in a Transformer-based RefineFormer Dosovitskiy (2020).

Formulation. We formulate the conditional flow-based refinement process using Conditional Flow Matching (CFM) Lipman et al. (2022); Tong et al. (2024). Instead of directly learning the flow velocity, we adopt the x-prediction parameterization to leverage the physics-anchored structure. The refinement trajectory is governed by the ordinary differential equation (ODE) starting from the physics-inspired prior:

$$\frac{d\mathcal{Z}_\tau}{d\tau} = u_\theta(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past}), \quad \text{with } \mathcal{Z}_0 = \hat{\mathcal{F}}_{phy}. \tag{12}$$

Here, the vector field u_θ is not predicted directly but is derived from the clean target predicted by the RefineFormer, denoted as net_θ:

$$u_\theta(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past}) = \frac{\text{net}_\theta(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past}) - \mathcal{Z}_\tau}{1-\tau}. \tag{13}$$

Training of PA-FM. During training, we optimize the RefineFormer to regress the vector field driving the physics-anchored trajectory. As shown in Algorithm 1, in each iteration, we sample a shared time step $\tau \sim p_\tau$ using a logit-normal-based schedule and construct the intermediate state $\mathcal{Z}_\tau$ via linear interpolation:

$$\mathcal{Z}_\tau = \tau \cdot \mathcal{F}_{GT} + (1-\tau) \cdot \hat{\mathcal{F}}_{phy}. \tag{14}$$

This linear path defines a constant target velocity field $u_{target} = \mathcal{F}_{GT} - \hat{\mathcal{F}}_{phy}$. The network net_θ takes $(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past})$ as input and implicitly defines the estimated vector field u_θ as per the x-prediction formulation derived above. Consequently, the training objective is to minimize the adaptive robust L_2 loss between the predicted and target vector fields:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{\tau, \mathcal{Z}_\tau, \mathcal{F}_{GT}} \left[\left(\| u_\theta(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past}) - u_{target} \|^2 + \epsilon \right)^\gamma \right], \tag{15}$$

where ϵ is a stability constant and γ controls the robustness to outliers. This objective corresponds to the v-loss illustrated in Figure 2.

Inference of PA-FM. We reconstruct a fine-grained prediction from a coarse-grained one by numerically solving the learned ODE initialized at the physics-inspired prior $\hat{Z}_0 = \hat{\mathcal{F}}_{phy}$. Using the Euler method with step size $\Delta\tau$, we integrate the trajectory from $\tau = 0$ to 1. As shown in Algorithm 2, no noise injection or random sampling is involved in the inference process. At each step i, the RefineFormer predicts the clean target $x_{\tau_i}^{pred} = \text{net}_\theta(\hat{Z}_{\tau_i}, \tau_i, \mathcal{F}_{past})$ to determine the local vector field. The state is updated via:

$$\hat{\mathcal{Z}}_{\tau_{i+1}} = \hat{\mathcal{Z}}_{\tau_i} + \Delta\tau \cdot \frac{x_{\tau_i}^{pred} - \hat{\mathcal{Z}}_{\tau_i}}{1 - \tau_i}. \tag{16}$$

The final integrated state $\hat{Z}_1$ serves as the refined prediction $\hat{\mathcal{F}}_{pred}$. Consequently, given a fixed input sequence, model parameters, number of integration steps N, and numerical solver, PA-FM produces a unique deterministic output.

4 Datasets and Evaluation Metrics

4.1 Datasets

We evaluate CloudFlow on Himawari-8/9 (AHI) and GOES-16 (ABI) datasets to assess its robustness across different monsoon and convective regimes. Both comprise 8 cloud-sensitive bands across visible, near-infrared, and infrared bands. Each sequence contains 12 frames at 30-min intervals: 6 past frames and 6 future frames at 256×256 resolution. **Himawari-8/9** dataset includes 5,032 sequences from 2022 for training and validation, and 865 from 2023 for testing. **GOES-16** dataset includes 6,719 sequences from 2020 for training and validation, and 1,088 from 2021 for testing. To support the evaluation of cross-region generalization, we additionally construct a test set from Himawari-8/9 observations covering northwestern China, a geographically distinct region from the original study area. This cross-region test set follows the same temporal and spectral settings and preprocessing pipeline as the original dataset and is reserved exclusively for evaluation. Further details such as band selection and cropping regions are provided in Appendix A.

4.2 Evaluation Metrics

Prediction performance is evaluated with MSE and MAE for pixel errors, PSNR and SSIM Wang et al. (2004) for reconstruction quality, and FVD Unterthiner et al. (2018) for perceptual realism and temporal coherence. For ensemble forecasts, CRPS is additionally reported as a proper score for predictive distributions Gneiting & Raftery (2007). Definitions are provided in Appendix C.

5 Results

5.1 Comparison with State-of-the-art Models

Experimental Setup. We conduct a comprehensive comparison by evaluating our CloudFlow with two categories of state-of-the-art (SOTA) models: deterministic and generative approaches. *1) Deterministic Models:* This category is further divided based on their structural design. The recurrent-based models include ConvLSTM Shi et al. (2015), PredRNN Wang et al. (2022), PhyDNet Guen & Thome (2020), and EPM Xia et al. (2024), which utilize recurrent units to capture temporal dependencies. In contrast, the non-recurrent models, such as Rainformer Bai et al. (2022), Earthformer Gao et al. (2022b), SimVP Gao et al. (2022a), Earthfarseer Wu et al. (2024), SimVPv2 Tan et al. (2025), and AlphaPre Lin et al. (2025), rely on convolution-based or transformer-based architectures for spatiotemporal modeling. *2) Generative Models:* This category includes the GAN-based model DGMR Ravuri et al. (2021) and two diffusion-based models. Specifically, MCVD Voleti et al. (2022) is a pixel space diffusion model, while PreDiff Gao et al. (2023) is a latent-space diffusion model. The detailed training configuration of CloudFlow is provided in Appendix D. All experiments are conducted on a single NVIDIA RTX 5090D GPU.

Quantitative Comparison. As shown in Table 1, CloudFlow achieves strong overall outperformance on both Himawari-8/9 and GOES-16 datasets. On Himawari-8/9, CloudFlow obtains the lowest

Table 1: Quantitative comparison with SOTA methods on Himawari-8/9 and GOES-16 datasets. Models are categorized into deterministic and generative approaches. Results are averaged across all spectral bands and lead times. For generative models, we report the mean over 10 independently seeded samples per input. Best results are highlighted in **bold**, second best in underlined. GFLOPs are provided in Appendix F.4, while the probabilistic evaluation and ensemble-size convergence analysis of the generative baselines are provided in Appendix F.2.

	Himawari-8/9					**GOES-16**				
Model	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
Deterministic Models										
ConvLSTM	61.33	4.570	23.43	0.655	482.2	90.48	5.058	23.97	0.714	249.1
PredRNN	74.28	5.153	22.90	0.659	467.7	91.70	5.125	23.92	0.712	248.4
PhyDNet	78.57	5.269	22.66	0.641	417.6	88.62	5.497	23.90	0.699	252.3
Rainformer	61.13	4.475	23.49	0.643	330.3	77.56	4.823	24.55	0.725	183.5
Earthformer	79.73	5.327	22.81	0.669	488.7	93.90	5.408	23.84	0.727	247.9
SimVP	54.25	4.171	23.71	0.637	264.4	85.26	4.962	24.22	0.719	182.1
Earthfarseer	72.74	5.089	22.94	0.632	295.5	93.62	5.386	23.74	0.696	216.0
EPM	66.33	4.715	23.42	0.653	434.4	81.14	4.812	24.39	0.711	212.3
SimVPv2	75.49	5.280	22.91	0.664	386.4	75.21	4.725	24.73	0.723	186.4
AlphaPre	56.45	4.266	23.78	**0.676**	294.7	81.61	5.002	24.42	0.755	161.2
Generative Models ($K = 10$ Ensemble Forecasts)										
MCVD	496.6	16.24	12.76	0.425	734.7	1274	24.33	12.13	0.457	534.7
DGMR	58.91	4.247	23.42	0.650	509.0	80.64	5.211	24.49	0.749	314.5
PreDiff	74.28	4.683	22.50	0.646	116.4	227.3	10.16	20.51	0.696	145.5
Ours										
CloudFlow	**53.87**	**4.129**	**23.79**	0.672	**67.56**	**74.02**	**4.179**	**24.74**	**0.760**	**51.66**

MSE of 53.87 and the lowest FVD of 67.56 among all evaluated methods, simultaneously achieving the best pixel-level accuracy and FVD-measured perceptual quality. CloudFlow also achieves the best reported MSE and FVD on GOES-16, further demonstrating consistent performance across datasets. A finer-grained comparison of MAE against deterministic models across individual spectral channels and forecast lead times is provided in Appendix F.1. Crucially, as detailed in Table 17, CloudFlow reduces GFLOPs by $\sim 812\times$ relative to PreDiff, while maintaining a computational cost in the same order of magnitude as deterministic baselines. These results demonstrate that CloudFlow combines accurate prediction, strong perceptual quality, and computational efficiency.

Qualitative Comparison. Figures 3, 18, 19, 20 and 21 visualize predictions from CloudFlow and other SOTA methods. Compared with deterministic models, CloudFlow recovers sharper details, whereas many baselines produce over-smoothed outputs. It also exhibits greater stability and structural integrity than diffusion-based models: CloudFlow mitigates the severe mean shifts and erratic pixel jumps observed in MCVD, maintaining temporally continuous intensity distributions, and preserves more coherent and complete cloud structures than PreDiff, avoiding spatial fragmentation. These improvements in clarity and consistency arise from deterministic iterative residual refinement rather than stochastic sampling. Overall, CloudFlow produces realistic spatiotemporal structures and outperforms the evaluated SOTA methods qualitatively.

5.2 Analysis

Interpretability. CloudFlow makes physical evolution explicit through its Evolution Network, which decomposes cloud evolution into motion-induced transport and intensity changes. Figures 13, 14, and 15 show a spatially coherent velocity field $\mathbf{v}$ aligned with observed cloud movement, while the intensity-change field $\mathbf{S}$ concentrates around cloud boundaries and evolving cloud bodies, indicating where clouds move and where their intensity develops or dissipates. Figure 12 further shows that 36.5% of velocities fall below the 0.56 m/s quiescence threshold; active-region speeds average

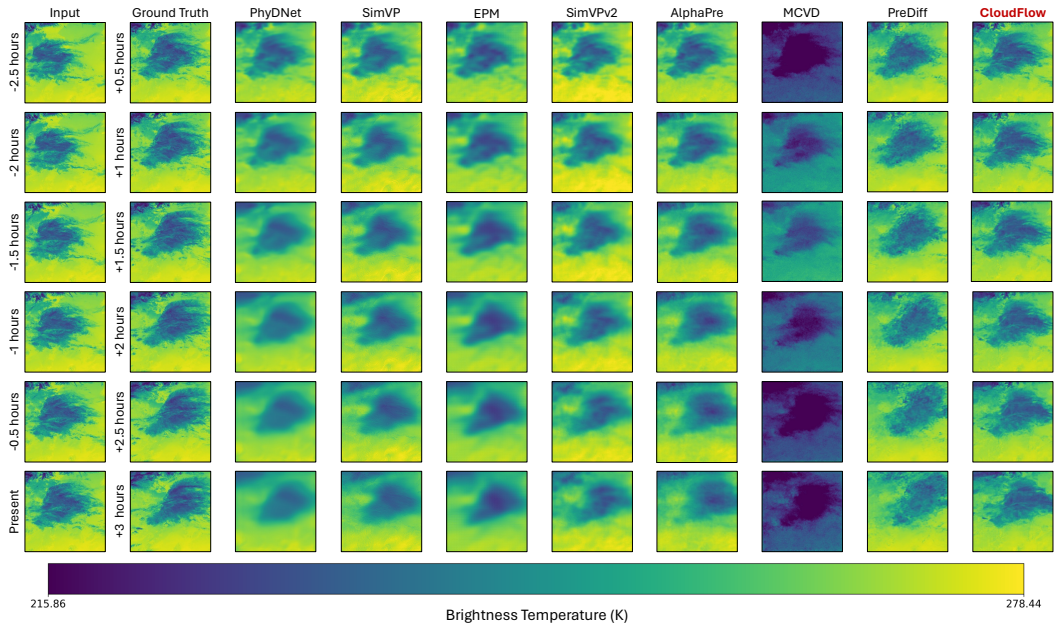


Figure 3: The 13.3 μm prediction result of different SOTA models on Southeast Asian coastal regions at 04/13/2023 05:40 (UTC). The first column shows the previous 6 timesteps of GMS data used as input. The second column is the ground truth GMS data we want to predict, and the subsequent columns display predictions from various baselines and our CloudFlow.

9.37 m/s and reach 64.8 m/s, consistent with reported Himawari-8 atmospheric motion vectors Otsuka et al. (2018); Santek et al. (2019).

Design Choices. Table 11 shows that $N = 10$ provides a favorable perception–accuracy trade-off, substantially improving FVD over $N = 5$ without the pronounced accuracy degradation observed at larger step counts. Table 12 shows that scale-aware DFI provides the most balanced fusion performance by matching dynamics features to the corresponding decoder scales. Table 13 confirms that $\alpha = 0.01$ effectively balances transport and source–sink corrections. Table 14 further shows that joint $6 \rightarrow 6$ prediction reduces six-step average MSE by 10.3% relative to autoregressive forecasting, supporting our multi-frame prediction design. Detailed descriptions are provided in Appendix E.

Generalization & Robustness. We conduct a cross-region experiment by transferring the checkpoints trained on the original Himawari-8/9 region directly to the cross-region dataset for evaluation, without retraining or fine-tuning. Table 18 shows that CloudFlow achieves the best MSE (75.47), MAE (4.880), SSIM (0.622), and FVD (77.06). Its FVD is 60.2% lower than the second-best result from PreDiff (193.8), demonstrating strong transfer across distinct terrain and climate conditions. We further examine robustness under extreme weather using Typhoon Mawar. Figures 4 and 21 show that CloudFlow preserves clear typhoon structures over a 3-hour horizon, whereas the baselines exhibit substantial blurring and structural degradation. Figure 22 shows stable performance across spring, autumn, and winter, with rapidly evolving summer convection remaining more challenging. Table 19 demonstrates controlled degradation under both additive and multiplicative noise, with greater tolerance to multiplicative

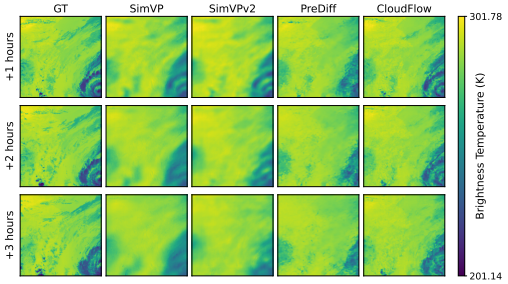


Figure 4: Predictions of CloudFlow and representative baselines for Typhoon Mawar in the 12.3 μm infrared band.

Table 2: Ablation study of Evolution Network components on the Himawari-8/9 dataset. SSD: Source-Sink Decoder, MD: Motion Decoder, GI: Grad Input, DFI: Dynamics Feature Integration.

SSD	MD	GI	DFI	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
✓				53.73	4.076	23.91	0.682	315.1
✓	✓			47.41	3.526	24.26	0.685	140.3
✓	✓	✓		45.20	3.319	24.37	0.699	**117.7**
✓	✓	✓	✓	**44.55**	**3.296**	**24.42**	**0.701**	130.4

Table 3: Ablation study of PA-FM on the Himawari-8/9 dataset. $\mathbf{x_0}$ denotes the prior distribution and $\mathbf{c}$ denotes the condition.

No.	$\mathbf{x_0}$	$\mathbf{c}$	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
I	$\mathcal{N}(0, \mathbf{I})$	$\mathcal{F}_{past}$	79.50	5.053	22.53	0.640	78.49
II	$\mathcal{N}(0, \mathbf{I})$	$\hat{\mathcal{F}}_{phy}$	65.89	4.541	23.32	0.662	**65.41**
III	$\mathcal{N}(0, \mathbf{I})$	$\mathcal{F}_{past}, \hat{\mathcal{F}}_{phy}$	71.63	4.813	23.05	0.656	84.31
IV	$\mathcal{F}_{past}$	$\hat{\mathcal{F}}_{phy}$	78.20	5.062	22.26	0.624	110.3
V	$\hat{\mathcal{F}}_{phy}$	$\mathcal{F}_{past}$	**53.87**	**4.129**	**23.79**	**0.672**	67.56

perturbations. Table 20 shows that CloudFlow retains useful predictive capability when earlier observations are missing; although losing the latest frame is more damaging, optical-flow replacement substantially restores performance. Detailed descriptions are provided in Appendix G.

5.3 Ablation Studies

Ablation for Evolution Network. Table 2 validates the effectiveness of each component. First, integrating Motion Decoder (**MD**) with Source-Sink Decoder (**SSD**) realizes the complete advection process, significantly reducing MSE and MAE by approximately 11.8% and 13.5%. Next, injecting Gradient Input (**GI**) into the encoder also increases prediction accuracy, reducing MSE and MAE by approximately 4.7% and 5.9%. Finally, Dynamics Feature Integration (**DFI**) yields consistent improvements across all MSE, MAE, PSNR and SSIM by conditioning intensity evolution on motion priors, though it slightly increases FVD as the enhanced physical constraints prioritize structural alignment over stochastic texture diversity.

Ablation for PA-FM. As shown in Table 3, we investigate different strategies for utilizing the physics-inspired prior $\hat{\mathcal{F}}_{phy}$ produced by the Evolution Network. **1) Conditioning (I–III).** Results indicate that II outperforms III and I across all metrics. This suggests that explicit physics-inspired priors $\hat{\mathcal{F}}_{phy}$ provide more effective guidance than implicit historical cues $\mathcal{F}_{past}$, and that combining both under noise-based settings (III) may introduce redundant information that hinders optimization. **2) Initialization (IV-V).** V significantly outperforms IV. Unlike IV, which forces the model to learn large-displacement advection from historical frames, V initializes near the target manifold using $\hat{\mathcal{F}}_{phy}$, simplifying the task to refinement. **3) Distribution vs. Condition (II vs. V).** Finally, V achieves optimal fidelity over II. By employing $\hat{\mathcal{F}}_{phy}$ as the starting distribution rather than just a condition, V avoids *de novo* construction from noise. Among the evaluated variants, V achieves the best MSE, MAE, PSNR, and SSIM, while II obtains a slightly lower FVD.

6 Conclusion

In this paper, we propose CloudFlow, a deterministic framework that combines a physics-inspired evolution prior with conditional flow-based refinement for multispectral geostationary satellite data nowcasting. CloudFlow models multi-scale atmospheric dynamics through the Evolution Network and uses Physics-Anchored Flow Matching to iteratively refine coarse predictions with localized details. By leveraging structured evolution priors and deterministic conditional refinement, CloudFlow produces accurate and visually sharp predictions, effectively alleviating the perception–accuracy trade-off. Comprehensive experiments on Himawari-8/9 and GOES-16 are executed from three aspects: (1) **Prediction Quality and Efficiency** compares accuracy and perceptual quality, measures computational efficiency, and evaluates generative baselines through CRPS and ensemble-size convergence analyses; (2) **Interpretability and Design Choices** analyses the learned motion and source–sink fields and validates the selected model components, DFI fusion strategy, α initialization, inference step count N, and joint multi-frame prediction scheme against autoregressive forecasting; and (3) **Generalization and Robustness** examines the model's generalization to a geographically distinct region, its performance under extreme-weather and seasonal conditions, its robustness to additive and multiplicative Gaussian noise, and its resilience to missing input frames. Future work will integrate low-latency thermodynamic and microphysical observations into a hybrid physical-neural source-sink module and extend the framework to long-term global nowcasting.

AI USE STATEMENT

In this work, we used generative AI tools to polish author-written text for grammatical correctness, clarity, and readability, and to assist with minor presentational editing of the manuscript. Generative AI was not used to originate the research questions or hypotheses, develop the conceptual or theoretical framework, design or implement the methodology or experiments, generate or process datasets, formulate mathematical claims or proofs, translate the manuscript, or perform scientific analysis or interpretation of the results. All technical content, claims, experimental results, and citations were produced or selected by the authors. The authors reviewed every AI-assisted revision against the original text to ensure that it preserved the intended technical meaning and introduced no unsupported claims or citations. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Cong Bai, Feng Sun, Jinglin Zhang, Yi Song, and Shengyong Chen. Rainformer: Features extraction balanced network for radar-based precipitation nowcasting. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

Kotaro Bessho, Kenji Date, Masahiro Hayashi, Akio Ikeda, Takahito Imai, Hidekazu Inoue, Yukihiro Kumagai, Takuya Miyakawa, Hidehiko Murata, Tomoo Ohno, et al. An introduction to himawari-8/9—japan's new-generation geostationary meteorological satellites. *Journal of the Meteorological Society of Japan. Ser. II*, 94(2):151–183, 2016.

Sandrine Bony, Hauke Schulz, Jessica Vial, and Bjorn Stevens. Sugar, gravel, fish, and flowers: Dependence of mesoscale patterns of trade-wind clouds on environmental conditions. *Geophysical research letters*, 47(7):e2019GL085988, 2020.

Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36:42038–42063, 2023.

Noah S Diffenbaugh, Deepti Singh, Justin S Mankin, Daniel E Horton, Daniel L Swain, Danielle Touma, Allison Charland, Yunjie Liu, Matz Haugen, Michael Tsiang, et al. Quantifying the influence of global warming on unprecedented extreme climate events. *Proceedings of the National Academy of Sciences*, 114(19):4881–4886, 2017.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3170–3180, 2022a.

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

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Bernie Wang. Prediff: Precipitation nowcasting with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:78621–78656, 2023.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

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

James R Holton and Gregory J Hakim. *An introduction to dynamic meteorology*, volume 88. Academic press, 2013.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Zhao-Liang Li, Bo-Hui Tang, Hua Wu, Huazhong Ren, Guangjian Yan, Zhengming Wan, Isabel F Trigo, and José A Sobrino. Satellite-derived land surface temperature: Current status and perspectives. *Remote sensing of environment*, 131:14–37, 2013.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17841–17850, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

D Merk and T Zinner. Detection of convective initiation using meteosat seviri: Implementation in and verification with the tracking and nowcasting algorithm cb-tram. *Atmospheric Measurement Techniques*, 6(8):1903–1918, 2013.

Michiko Otsuka, Hiromu Seko, Kazuki Shimoji, and Koji Yamashita. Characteristics of Himawari-8 rapid scan atmospheric motion vectors utilized in mesoscale data assimilation. *Journal of the Meteorological Society of Japan. Ser. II*, 96B:111–131, 2018. doi: 10.2151/jmsj.2018-034.

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.

David Santek, Richard Dworak, Sharon Nebuda, Steve Wanzong, Régis Borde, Iliana Genkova, Javier García-Pereda, Renato Galante Negri, Manuel Carranza, Kenichi Nonaka, Kazuki Shimoji, Soo Min Oh, Byung-Il Lee, Sung-Rae Chung, Jaime Daniels, and Wayne Bresky. 2018 atmospheric motion vector (AMV) intercomparison study. *Remote Sensing*, 11(19):2240, 2019. doi: 10.3390/rs11192240.

Timothy J Schmit, Paul Griffith, Mathew M Gunshor, Jaime M Daniels, Steven J Goodman, and William J Lebair. A closer look at the abi on the goes-r series. *Bulletin of the American Meteorological Society*, 98(4):681–698, 2017.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=CD9Snc73AW.

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

Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. *arXiv preprint arXiv:2404.10024*, 2024.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

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

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

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

Pan Xia, Lu Zhang, Min Min, Jun Li, Yun Wang, Yu Yu, and Shengjie Jia. Accurate nowcasting of cloud cover at solar photovoltaic plants using geostationary satellite images. *Nature Communications*, 15(1):510, 2024.

Xin Zhang, Ya'nan Zhou, and Jiancheng Luo. Deep learning for processing and analysis of remote sensing big data: A technical review. *Big Earth Data*, 6(4):527–560, 2022.

APPENDIX

A DETAILED DATASET DESCRIPTION

This section describes the Himawari-8/9 AHI and GOES-16 ABI datasets used for model development and evaluation, together with an additional Himawari-8/9 dataset for cross-region testing. Figure 5 summarizes the geographic coverage, terrain, and climate of the three study regions.

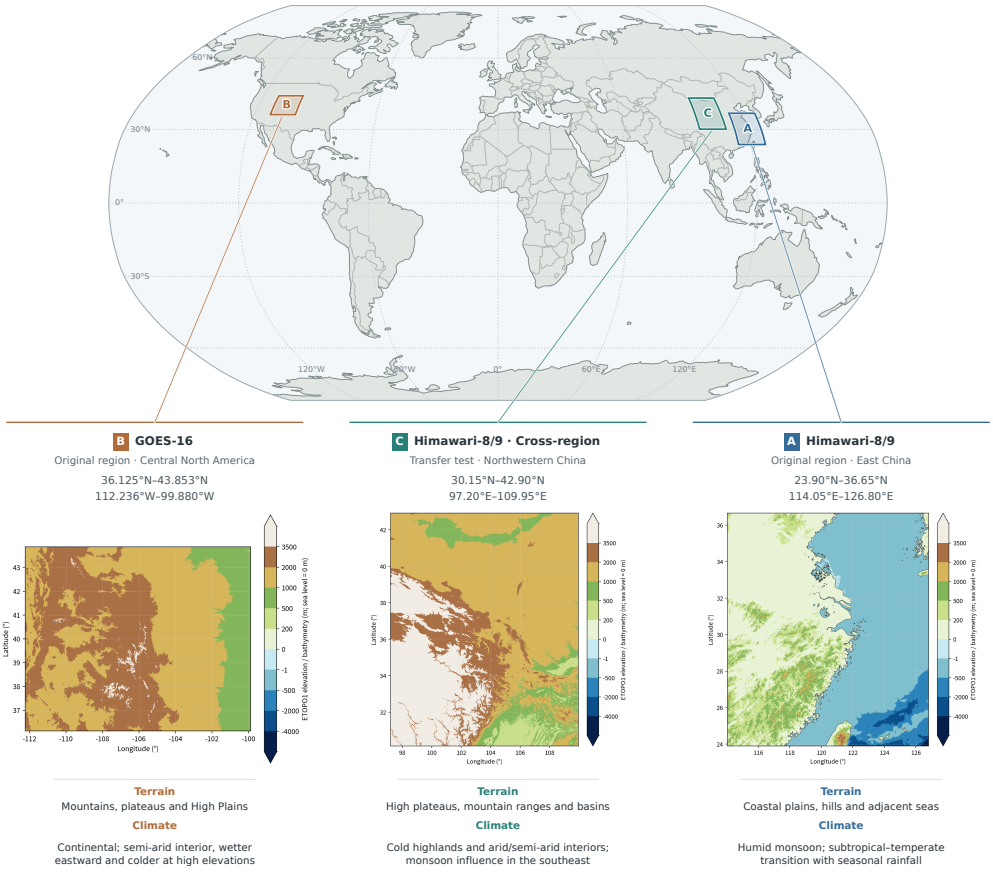


Figure 5: Geographic coverage and environmental characteristics of the three study regions: (A) the original Himawari-8/9 region in East China and adjacent seas; (B) the GOES-16 region in central North America; and (C) the Himawari-8/9 cross-region test area in northwestern China. Colored outlines and matching connections identify the corresponding terrain maps, with elevation and bathymetry in meters relative to sea level.The latitude-longitude bounds of each study region are shown alongside its terrain map, together with a concise description of the region's topography and climate.

A.1 HIMAWARI-8/9 AHI DATASET

The first dataset is derived from the Advanced Himawari Imager (AHI) onboard the Himawari-8/9 geostationary satellites.

- **Spatial Extent:** The study area is cropped to 23.90°N – 36.65°N and 114.05°E – 126.80°E, covering East China and adjacent sea areas.
- **Resolution:** The original data features a spatial resolution of 4 km and a temporal resolution of 10 minutes.

- **Temporal Coverage (Daytime Only):** To leverage reflective visible spectral information for fine-grained cloud texture analysis, we specifically select **daytime data** from **01:00 to 09:00 UTC**.
- **Sample Statistics:** Using overlapping sliding windows for training and non-overlapping windows for testing, the dataset contains **5,032** sequences from the year 2022 for training and validation, and **865** sequences from 2023 for independent testing. Each sequence consists of 12 frames at 30-minute intervals.

The selected eight bands and their corresponding meteorological semantic information are summarized in Table 4. A visualization of a typical sequence is provided in Figure 6.

Table 4: Spectral band configurations and semantic information for the Himawari-8/9 AHI dataset.

Band	Center λ (μm)	Type	Meteorological Application/Semantics
B03	0.64	Visible	Day-time cloud texture and fog detection
B05	1.6	Near-IR	Cloud phase (ice vs. water particles) discrimination
B07	3.9	Short-wave IR	Night-time cloud tracking and fire/hotspot detection
B11	8.6	Long-wave IR	Cloud phase monitoring and SO_2 concentration
B13	10.4	Long-wave IR	Clean IR window for surface and cloud top temperature
B14	11.2	Long-wave IR	Core band for deep convection and cloud height estimation
B15	12.3	Long-wave IR	Dirty IR window for low-level moisture and dust tracking
B16	13.3	Long-wave IR	CO_2 absorption band for cloud top pressure estimation

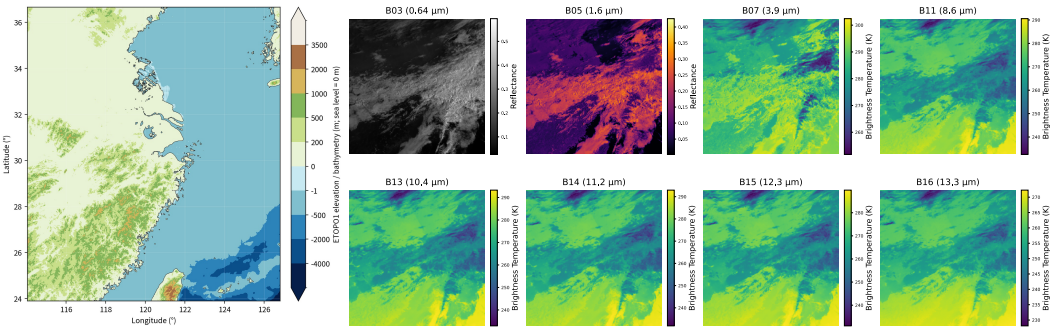


Figure 6: Visualization of the **Himawari-8/9 AHI** study region, including its terrain elevation and representative multispectral samples from the eight selected bands.

A.2 GOES-16 ABI Dataset

The second dataset utilizes the Advanced Baseline Imager (ABI) onboard the GOES-16 satellite, providing a different geographical and resolution perspective.

- **Spatial Extent:** The area is cropped to 36.125°N – 43.853°N and 112.236°W – 99.880°W. This region is located in the central interior of North America, primarily covering a vast portion of the Great Plains.
- **Resolution:** This dataset offers a higher spatial resolution of 2 km and a temporal resolution of 5 minutes.
- **Temporal Coverage (Daytime Only):** Similar to the AHI dataset, we extract **daytime data** from **14:31 to 23:31 UTC** daily to ensure the consistent availability of high-resolution reflective channels.
- **Sample Statistics:** The dataset comprises **6,719** training/validation sequences from 2020 and **1,088** test sequences from 2021.

The analogous spectral channels selected for GOES-16 are detailed in Table 5. The multi-band evolution for this dataset is visualized in Figure 7.

Table 5: Spectral band configurations and semantic information for the GOES-16 ABI dataset.

Band	**Center λ (μm)**	**Type**	**Meteorological Application/Semantics**
C02	0.64	Visible	High-resolution daytime cloud and insolation monitoring
C05	1.61	Near-IR	Snow/cloud discrimination and particle size estimation
C07	3.90	Short-wave IR	Identification of fog, fire, and nocturnal cloud features
C12	9.61	Long-wave IR	Total column ozone and upper-level dynamics
C13	10.35	Long-wave IR	Clean IR window for surface and cloud temperature
C14	11.2	Long-wave IR	Convective cloud growth and intensity monitoring
C15	12.3	Long-wave IR	Low-level moisture tracking (Split-window difference)
C16	13.3	Long-wave IR	CO_2 absorption for cloud height and atmospheric stability

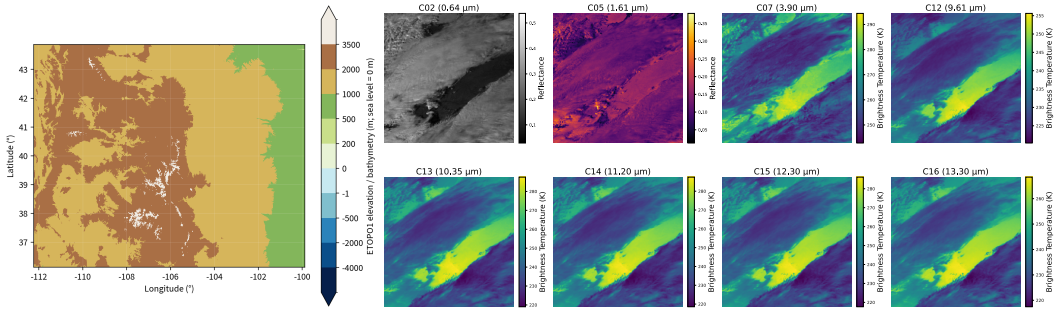


Figure 7: Visualization of the **GOES-16 ABI** study region, including its terrain elevation and representative multispectral samples from the eight selected bands.

A.3 Cross-Region Dataset

We assess geographic transfer using an additional Himawari-8/9 test region in northwestern China (region C in Figure 5), covering 30.15°N–42.90°N and 97.20°E–109.95°E.

Apart from the geographic coverage, the cross-region dataset follows the same temporal and spectral settings as the original Himawari-8/9 test set: observations are taken from 2023 within the same daytime window of 01:00–09:00 UTC, using the identical eight AHI bands listed in Table 4. The sequence configuration is also unchanged, with six historical frames used to predict six future frames at 30-minute intervals.

The domain spans the northeastern Tibetan Plateau, the Qilian Mountains, the Hexi Corridor, the Loess Plateau, and surrounding inland basins and plains. Compared with the original coastal and plain-dominated region, it has larger elevation gradients and more heterogeneous terrain. Its climate ranges from cold high-elevation conditions to arid and semi-arid continental regimes, with stronger East Asian summer monsoon influence toward the southeast.

This additional dataset is used for cross-region testing rather than model training. For every evaluated method, we directly apply the checkpoint trained on the original Himawari-8/9 region without retraining or fine-tuning. The resulting comparisons are reported in Appendix G.1 and Table 18.

A.4 Preprocessing Details

For both datasets, each frame is cropped to a spatial size of 256×256 pixels. The nowcasting task is defined as using a sequence of six historical frames to predict the subsequent six frames at 30-minute sampling intervals. All data are normalized to a range of $[0, 1]$ based on the maximum and minimum values of each spectral band observed in the training set.

B Model Details

B.1 Details of Gradient-Enhanced Encoder

The Gradient-Enhanced Encoder acts as the primary feature extractor, transforming raw satellite radiances into a multi-scale latent representation. It incorporates a gradient injection mechanism to explicitly preserve high-frequency structural boundaries (e.g., cloud fronts) that are often lost in standard downsampling operations.

Gradient Injection. Let the input sequence be denoted as $\mathbf{X} \in \mathbb{R}^{T_{in} \times C \times H \times W}$. We augment this input with spatial gradient magnitudes to provide a strong structural prior. The gradients $\mathbf{G}$ are computed via a fixed Sobel operator $\mathcal{S}$:

$$\mathbf{G} = \mathcal{S}(\mathbf{X}) = \sqrt{(\mathbf{X} * \mathbf{K}_x)^2 + (\mathbf{X} * \mathbf{K}_y)^2}, \tag{17}$$

where $\mathbf{K}_x, \mathbf{K}_y$ are the 3×3 Sobel kernels. The network input $\mathbf{X}_{in}$ is the channel-wise concatenation:

$$\mathbf{X}_{in} = \text{Concat}(\mathbf{X}, \mathbf{G}) \in \mathbb{R}^{T_{in} \times 2C \times H \times W}. \tag{18}$$

Hierarchical Encoding. The encoder employs a 4-stage hierarchy. Each stage l consists of a Residual Block (ResBlock) followed by 2×2 Max-Pooling. The feature propagation is formalized as:

$$\begin{aligned} \mathbf{F}_{up}^{(l)} &= \text{ConvTranspose}(\mathbf{F}_{mot}^{(l+1)}), \\ \mathbf{F}_{att}^{(l)} &= \text{AttnGate}(\mathbf{F}_{enc}^{(l)}, \mathbf{F}_{up}^{(l)}), \\ \mathbf{F}_{mot}^{(l)} &= \text{ConvBlock}(\text{Concat}[\mathbf{F}_{att}^{(l)}, \mathbf{F}_{up}^{(l)}]). \end{aligned} \tag{19}$$

The final representation is compressed into a bottleneck tensor $\mathbf{F}_{btl} = \text{ResBlock}_{btl}(\mathbf{F}_{enc}^{(4)})$, which encodes the global spatiotemporal context used by both downstream decoders.

Table 6: Architecture of Gradient-Enhanced Encoder. $C_{base} = 32$.

Stage	Component	Kernel	Channels	Output Size
Input	Sobel + Concat	-	$2C \cdot T_{in}$	$H \times W$
Stage 1	ResBlock + Pool	3×3	C_{base}	$H/2 \times W/2$
Stage 2	ResBlock + Pool	3×3	$2C_{base}$	$H/4 \times W/4$
Stage 3	ResBlock + Pool	3×3	$4C_{base}$	$H/8 \times W/8$
Stage 4	ResBlock + Pool	3×3	$8C_{base}$	$H/16 \times W/16$
Bottleneck	ResBlock	3×3	$16C_{base}$	$H/16 \times W/16$

B.2 Details of Motion Decoder

To disentangle large-scale advection from local deformation, we decompose the total velocity field prediction $\mathbf{V} \in \mathbb{R}^{T_{out} \times 2 \times H \times W}$ into a global rigid component and a local residual:

$$\mathbf{V} = \mathbf{V}_{global}(\mathbf{F}_{btl}) + \Delta\mathbf{V}_{local}(\mathbf{F}_{btl}, \{\mathbf{F}_{enc}^{(l)}\}). \tag{20}$$

Global Rigid Flow Head. A lightweight Multi-Layer Perceptron (MLP) projects the pooled bottleneck features $\mathbf{F}_{btl}$ into five rigid-motion parameters $\{\theta, c_x, c_y, d_x, d_y\}$. A coordinate grid generator then constructs the coarse flow field $\mathbf{V}_{global}$ corresponding to the background translation and rotation.

Attention-Gated Local Decoder. The local branch predicts the non-rigid residual flow $\Delta\mathbf{V}$. It employs a U-Net decoder with **Attention Gates (AGs)**. AGs filter the skip connections from the encoder ($\mathbf{F}_{enc}^{(l)}$) using the upsampled decoder features ($\mathbf{F}_{mot}^{(l+1)}$) as gating signals, suppressing background noise. The decoding operation at stage l is defined as:

$$\begin{aligned} \mathbf{F}_{up}^{(l)} &= \text{ConvTranspose}(\mathbf{F}_{mot}^{(l+1)}), \\ \mathbf{F}_{att}^{(l)} &= \text{AttnGate}(\mathbf{F}_{enc}^{(l)}, \mathbf{F}_{up}^{(l)}), \\ \mathbf{F}_{mot}^{(l)} &= \text{ConvBlock}(\text{Concat}[\mathbf{F}_{att}^{(l)}, \mathbf{F}_{up}^{(l)}]). \end{aligned} \tag{21}$$

A final convolution layer maps $\mathbf{F}_{mot}^{(1)}$ to $\Delta\mathbf{V}$.

Table 7: Configuration of Motion Decoder. Note: 'AttnGate' filters the skip connection $\mathbf{F}_{enc}^{(l)}$.

Stage	Input	Component	Channels
Global	$\mathbf{F}_{btl}$	Pooling $\rightarrow$ MLP	5 (Rigid)
Dec 4	$\mathbf{F}_{btl}$	Up + AttnGate w/ $\mathbf{F}_{enc}^{(4)}$	$8C_{base}$
Dec 3	$\mathbf{F}_{mot}^{(4)}$	Up + AttnGate w/ $\mathbf{F}_{enc}^{(3)}$	$4C_{base}$
Dec 2	$\mathbf{F}_{mot}^{(3)}$	Up + AttnGate w/ $\mathbf{F}_{enc}^{(2)}$	$2C_{base}$
Dec 1	$\mathbf{F}_{mot}^{(2)}$	Up + AttnGate w/ $\mathbf{F}_{enc}^{(1)}$	C_{base}
Head	$\mathbf{F}_{mot}^{(1)}$	1×1 Conv	$2 \cdot C \cdot T_{out}$

B.3 Details of Source-Sink Decoder

To model non-conservative processes (e.g., convection, dissipation), this branch predicts the intensity residual field $\mathbf{S} \in \mathbb{R}^{T_{out} \times C \times H \times W}$. This decoder is explicitly conditioned on the fluid dynamics derived from the predicted velocity $\mathbf{V}$.

Differentiable Physics Extraction. For a predicted velocity vector field $\mathbf{v} = (u, v)$, we compute a set of kinematic feature maps $\mathbf{Z}_{dyn}$ using differentiable finite differences. We adopt a unified notation $\mathbf{z}_{\{\cdot\}}$ to represent these dynamics priors:

$$\begin{aligned} \text{Div: } \mathbf{z}_{\text{div}} &= u_x + v_y, & \text{Vort: } \mathbf{z}_{\text{vort}} &= v_x - u_y, \\ \text{Def 1: } \mathbf{z}_{e1} &= u_x - v_y, & \text{Def 2: } \mathbf{z}_{e2} &= v_x + u_y, \\ \text{Shear: } \mathbf{z}_{\text{shear}} &= \sqrt{\mathbf{z}_{e1}^2 + \mathbf{z}_{e2}^2}, & \text{Strain: } \mathbf{z}_{\text{strain}} &= \sqrt{\mathbf{z}_{\text{div}}^2 + \mathbf{z}_{\text{shear}}^2}. \end{aligned} \tag{22}$$

Additionally, the flow speed is denoted as $\mathbf{z}_{\text{speed}} = \|\mathbf{v}\|$. The full dynamics feature set is defined as $\mathbf{Z}_{dyn} = \{\mathbf{z}_{\text{div}}, \mathbf{z}_{\text{vort}}, \mathbf{z}_{e1}, \mathbf{z}_{e2}, \mathbf{z}_{\text{shear}}, \mathbf{z}_{\text{strain}}, \mathbf{z}_{\text{speed}}\}$.

Physics-Anchored Injection. We employ a scale-aware strategy to inject these physical features into the Source-Sink Decoder. We hypothesize that bulk properties (speed, shear) align with deep semantic features, while local properties (divergence) align with shallow spatial features.

Let $\mathbf{F}_{src}^{(l)}$ denote the feature maps of the Source-Sink Decoder at the l-th stage. The fusion update rule is formulated as:

$$\mathbf{F}_{src}^{(l)} = \text{ConvBlock}\left(\text{Concat}\left[\mathbf{F}_{up}^{(l)}, \mathbf{F}_{enc}^{(l)}, \mathcal{P}_l(\mathbf{Z}_{subset})\right]\right), \tag{23}$$

where $\mathbf{F}_{up}^{(l)}$ represents the upsampled features from the previous decoder layer, $\mathbf{F}_{enc}^{(l)}$ denotes the skip-connected features from the encoder, and $\mathcal{P}_l$ is the projection layer adapted for the specific subset of physical features $\mathbf{Z}_{subset}$. The output head generates the residual field: $\mathbf{S} = \text{Conv}(\mathbf{F}_{src}^{(1)})$.

Table 8: Physics Injection Hierarchy for Source-Sink Decoder.

Fusion Stage	Target Feature	Physical Features ($\mathbf{Z}_{subset}$)	Rationale
Early	$\mathbf{F}_{src}^{(4)}, \mathbf{F}_{src}^{(3)}$	$\mathbf{z}_{\text{speed}}, \mathbf{z}_{\text{shear}}, \mathbf{z}_{e1}, \mathbf{z}_{e2}$	Global kinematics
Mid	$\mathbf{F}_{src}^{(2)}$	$\mathbf{z}_{\text{vort}}$	Meso-scale rotation
Late	$\mathbf{F}_{src}^{(1)}$	$\mathbf{z}_{\text{div}}, \mathbf{z}_{\text{strain}}$	Local intensity source

B.4 Details of Semi-Lagrangian Extrapolation

In this section, we elaborate on the **semi-Lagrangian extrapolation scheme** adopted in our model and establish its discrete realization to our "Warp-and-Add" operation. The semi-Lagrangian method

solves evolution problems in the Eulerian framework by tracing the Lagrangian trajectories of fluid parcels, making it particularly suitable for advection-dominated spatiotemporal sequence prediction tasks.

Lagrangian Perspective of Evolution. Traditional Eulerian methods describe the rate of change of physical quantities at fixed grid points, which are prone to numerical dissipation and stability issues when handling high-speed fluid dynamics problems. The semi-Lagrangian method adopts a hybrid perspective: computations are performed on a fixed Eulerian grid, but physical quantities are updated by tracking the Lagrangian trajectories of fluid parcels.

For any physical quantity field $\mathbf{X}$ moving with the fluid, its temporal evolution can be described by the **material derivative**, which represents the rate of change observed while following a fluid parcel along its trajectory:

$$\frac{D\mathbf{X}}{Dt} = \mathbf{F}(\mathbf{X}, \mathbf{v}), \tag{24}$$

where $\mathbf{v}$ denotes the velocity field, and $\mathbf{F}$ represents the net forcing term acting on the fluid parcel, encompassing all non-transport effects such as thermodynamic processes, source-sink dynamics, and intensity variations caused by fluid expansion or compression.

semi-Lagrangian Numerical Solution Scheme. To solve Eq. 24 on a fixed regular grid and obtain the physical quantity field at the next time step $t+1$, the semi-Lagrangian method employs a **backward trajectory tracing** strategy, which consists of two core steps:

1. **Departure Point Estimation:** For each target grid point $\mathbf{p}$ at time $t+1$, we trace backward along the velocity field to find the origin (departure point) of the fluid parcel that will arrive at $\mathbf{p}$ at time $t+1$. Using a first-order Euler approximation with unit time step $\Delta t = 1$:

$$\mathbf{p}_{dep} \approx \mathbf{p} - \mathbf{v}(\mathbf{p}). \tag{25}$$

2. **Discrete Time Update:** We discretize the material derivative along the characteristic line, yielding the time update rule for physical quantities:

$$\mathbf{X}_{t+1}(\mathbf{p}) = \mathbf{X}_t(\mathbf{p}_{dep}) + \mathbf{F}(\mathbf{p}). \tag{26}$$

   This equation states that the value of the physical quantity at grid point $\mathbf{p}$ at time $t+1$ equals the value at the departure point $\mathbf{p}_{dep}$ at time t, plus the cumulative effect of all net forcing terms acting on the fluid parcel during its motion.

Differentiable Neural Implementation. In our neural network framework, we do not explicitly compute the contributions of individual physical processes separately. Instead, we train a **Source-Sink Decoder** to directly predict a unified residual field $\mathbf{S}$ that approximates the entire net forcing term $\mathbf{F}$ in Eq. 24. To enhance the model's expressive power, we introduce a learnable weight coefficient α to regulate the influence strength of the residual term.

This leads to the final update rule adopted in our model:

$$\mathbf{X}_{t+1}(\mathbf{p}) = \underbrace{\mathbf{X}_t(\mathbf{p} - \mathbf{v}(\mathbf{p}))}_{\text{Extrapolation Term } \mathcal{W}} + \alpha \mathbf{S}(\mathbf{p}). \tag{27}$$

The extrapolation term $\mathcal{W}$ in Eq. 27 requires evaluating the physical quantity field at non-integer coordinates $\mathbf{p}_{dep}$. We implement this via a differentiable bilinear interpolation operation (Grid Sample):

$$\mathcal{W}(\mathbf{X}_t, \mathbf{v})[\mathbf{p}] = \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p}_{dep})} \mathbf{X}_t(\mathbf{q}) \cdot \prod_{d \in \{x,y\}} \max(0, 1 - |(\mathbf{p}_{dep})_d - q_d|), \tag{28}$$

where $\mathcal{N}(\mathbf{p}_{dep})$ denotes the four integer grid points surrounding the departure point $\mathbf{p}_{dep}$.

The above derivation rigorously proves that our "Warp-and-Add" architecture is mathematically equivalent to a semi-Lagrangian solver. The learnable residual field $\mathbf{S}$ can adaptively compensate for all non-transport effects, including complex thermodynamic processes, fluid compression and expansion, and numerical discretization errors, thereby achieving strong data fitting capability.

B.5 Details of RefineFormer

This section provides the detailed implementation of the **RefineFormer** (net_θ), corresponding to the backbone architecture of our Physics-Anchored Flow Matching (PA-FM) framework. As introduced in the main text, RefineFormer is trained to predict the clean target state $\mathcal{F}_{GT}$ given the intermediate state $\mathcal{Z}_\tau$, the time step τ, and the historical context $\mathcal{F}_{past}$.

Input Representation and Tokenization. The network inputs are the current intermediate state $\mathcal{Z}_\tau \in \mathbb{R}^{T\times C\times H\times W}$ and the historical context $\mathcal{F}_{past} \in \mathbb{R}^{T\times C\times H\times W}$. Unlike standard video transformers that use 3D patch embedding, we employ a **Channel-Time Flattening** strategy to preserve temporal coherence without excessive computational cost. We concatenate the input state and the condition along the channel dimension to form a joint tensor, and fold the temporal dimension T into the channel space:

$$\mathbf{X}_{in} = \text{Concat}(\mathcal{Z}_\tau, \mathcal{F}_{past}) \in \mathbb{R}^{(T\cdot 2C)\times H\times W}. \tag{29}$$

This 2D representation allows the model to process temporal dynamics as channel interactions within the transformer blocks.

Bottleneck Patch Embedding. Mapping high-resolution meteorological data directly to the transformer's hidden dimension D is parameter-intensive. To mitigate this, we implement a **Bottleneck Patch Embedding** module. Instead of a single linear projection, we use a two-stage convolutional process:

1. **Compression**: A strided convolution with kernel size and stride $P = 16$ maps $\mathbf{X}_{in}$ to a low-dimensional bottleneck space $D_{btn} = 256$.
2. **Projection**: A pointwise 1×1 convolution projects the bottleneck features to the working hidden dimension $D = 768$.

Formally, the sequence of patch tokens $\mathbf{E} \in \mathbb{R}^{N\times D}$ (where $N = HW/P^2$) is obtained via:

$$\mathbf{E} = \text{Conv}_{1\times 1}\left(\sigma\left(\text{Conv}_{P\times P}(\mathbf{X}_{in})\right)\right) + \mathbf{P}_{pos}, \tag{30}$$

where $\mathbf{P}_{pos}$ are learnable positional embeddings.

Adaptive Transformer Backbone (AdaLN-Zero). The core of RefineFormer consists of $L = 12$ DiT-style transformer blocks. To enforce the ODE trajectory defined in Eq. 12, the network must be conditioned on the continuous time step τ. We utilize **Adaptive Layer Normalization (AdaLN)** to inject τ into every block. The time step τ is first mapped to a sinusoidal embedding and processed by an MLP to generate a conditioning vector $\mathbf{c}$. Within each block, $\mathbf{c}$ regresses the scale γ, shift β, and gating α parameters. The calculation for a hidden state $\mathbf{h}$ is:

$$\begin{aligned} \hat{\mathbf{h}} &= \text{MSA}(\text{Modulate}(\text{LN}(\mathbf{h}), \mathbf{c})), \\ \mathbf{h}_{mid} &= \mathbf{h} + \alpha_1(\mathbf{c}) \cdot \hat{\mathbf{h}}, \\ \mathbf{h}_{out} &= \mathbf{h}_{mid} + \alpha_2(\mathbf{c}) \cdot \text{MLP}(\text{Modulate}(\text{LN}(\mathbf{h}_{mid}), \mathbf{c})), \end{aligned} \tag{31}$$

where $\text{Modulate}(\mathbf{x}, \mathbf{c}) = \mathbf{x} \cdot (1 + \gamma(\mathbf{c})) + \beta(\mathbf{c})$. We employ the **zero-initialization** strategy for α_1, α_2, initializing the block as an identity function to stabilize the learning of the physics-anchored trajectory.

x-Prediction Head. In alignment with our PA-FM formulation, the network is parameterized to predict the clean target $\mathcal{F}_{GT}$ rather than the noise or velocity. The final latent representation is decoded via an AdaLN-modulated linear layer followed by an **Unpatchify** operation. This rearranges the patch tokens back into the spatial grid and reshapes the channel dimension to recover the predicted spatiotemporal state $\hat{\mathcal{F}} \in \mathbb{R}^{T\times C\times H\times W}$. This output is used to compute the vector field u_θ as shown in Eq. 13.

B.6 Implementation Details

The procedural details of PA-FM are summarized in Alg. 1 and 2. We implement the x-prediction parameterization as defined in Eq. (13), where the network net_θ regresses the target $\mathcal{F}_{GT}$ to implicitly

Algorithm 1 Training for PA-FM

Input: Ground truth $\mathcal{F}_{GT}$, physics-inspired prior $\hat{\mathcal{F}}_{phy}$, past context $\mathcal{F}_{past}$
repeat
  Sample $r \sim \mathcal{U}(0,1)$ and set $\tau \leftarrow \text{sigmoid}\big(\Phi^{-1}(\sqrt{r}) - 0.4\big)$ $\{\tau \sim p_\tau\}$
  $\mathcal{Z}_\tau \leftarrow \tau \cdot \mathcal{F}_{GT} + (1-\tau) \cdot \hat{\mathcal{F}}_{phy}$ {Linear path}
  $x_\tau^{pred} \leftarrow \text{net}_\theta(\mathcal{Z}_\tau, \tau, \mathcal{F}_{past})$ {x-prediction}
  $u_\theta \leftarrow (x_\tau^{pred} - \mathcal{Z}_\tau) / \max(1-\tau, \epsilon_{mask})$
  $u_{target} \leftarrow \mathcal{F}_{GT} - \hat{\mathcal{F}}_{phy}$
  Update θ by minimizing $\mathcal{L}_{CFM}(u_\theta, u_{target})$
until converged

Algorithm 2 Inference for PA-FM

Input: Physics-inspired prior $\hat{\mathcal{F}}_{phy}$, context $\mathcal{F}_{past}$, steps N
$\hat{\mathcal{Z}}_0 \leftarrow \hat{\mathcal{F}}_{phy}$, $\Delta\tau \leftarrow 1/N$ {Deterministic initialization; no stochastic sampling}
for $i = 0$ **to** $N-1$ **do**
  $\tau_i \leftarrow i/N$
  $x_{\tau_i}^{pred} \leftarrow \text{net}_\theta(\hat{\mathcal{Z}}_{\tau_i}, \tau_i, \mathcal{F}_{past})$
  $u_\theta \leftarrow (x_{\tau_i}^{pred} - \hat{\mathcal{Z}}_{\tau_i}) / (1-\tau_i)$
  $\hat{\mathcal{Z}}_{\tau_{i+1}} \leftarrow \hat{\mathcal{Z}}_{\tau_i} + u_\theta \cdot \Delta\tau$
end for
Return refined prediction $\hat{\mathcal{F}}_{pred} = \hat{\mathcal{Z}}_1$

derive the velocity field. During training, an intermediate state $\mathcal{Z}_\tau$ is constructed along the linear path between the physics-inspired prior $\hat{\mathcal{F}}_{phy}$ and the ground truth $\mathcal{F}_{GT}$ at a randomly sampled training time step τ, according to Eq. (14). Sampling τ only selects training points along this prescribed path and does not represent ensemble sampling. During inference, a discrete Euler solver iteratively integrates the learned vector field from the deterministic initialization $\hat{\mathcal{F}}_{phy}$, refining the coarse nowcasts into fine-grained results over N steps. **No noise injection or stochastic sampling is used in the default inference procedure.**

C Evaluation Metrics

Here, we present the quantitative metrics used to evaluate the comparative performance of different models. Let $Y \in \mathbb{R}^{T\times C\times H\times W}$ denote the ground-truth sequence and $\hat{Y}$ denote the predicted sequence, where T, C, H, and W represent time steps, channels, height, and width, respectively. MSE, MAE, and physical-space CRPS are computed after inverse normalization. PSNR, SSIM, and normalized CRPS are computed independently for each spectral band after applying the same band-specific transformation to predictions and targets to map them to $[0,1]$, and are then averaged across bands.

1. Accuracy Metrics

Mean Squared Error (MSE) measures the average squared deviation between predicted and ground-truth pixels in the denormalized physical space:

$$\text{MSE} = \frac{1}{TCHW}\sum_{t=1}^{T}\sum_{c=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(Y_{t,c,i,j} - \hat{Y}_{t,c,i,j}\right)^2. \tag{32}$$

Mean Absolute Error (MAE) measures the average absolute difference in the denormalized physical space:

$$\text{MAE} = \frac{1}{TCHW}\sum_{t=1}^{T}\sum_{c=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{W}\left|Y_{t,c,i,j} - \hat{Y}_{t,c,i,j}\right|. \tag{33}$$

2. Visual Quality Metrics

Peak Signal-to-Noise Ratio (PSNR) is computed separately for each band in the normalized $[0,1]$ space:

$$\text{PSNR}_c = 10\log_{10}\left(\frac{1}{\text{MSE}_c^{\text{norm}}}\right), \tag{34}$$

where $\text{MSE}_c^{\text{norm}}$ is the MSE for band c after band-specific normalization. The reported PSNR is averaged across bands.

Structural Similarity Index (SSIM) evaluates luminance, contrast, and structural similarity. For normalized images x and y, it is computed as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{35}$$

SSIM is computed independently for each band in the normalized $[0, 1]$ space and then averaged across bands.

3. Perceptual Quality Metric

Fréchet Video Distance (FVD) measures the distance between the distribution of real videos (P_r) and generated videos (P_g). It uses a pre-trained Inflated 3D ConvNet (I3D) to extract features:

$$\text{FVD} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}), \tag{36}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariance of the feature representations of the real and generated sequences, respectively. For multispectral evaluation, each spectral band is processed independently: its single-channel video is replicated three times to form a pseudo-RGB input to I3D. We compute FVD separately for all eight bands and report their unweighted average. For a fixed set of predictions and ground-truth sequences, FVD computation is deterministic. Therefore, the reported variance of FVD for generative baselines comes from stochastic generation under different random seeds, rather than from the metric itself.

4. Probabilistic Forecast Metrics

Continuous Ranked Probability Score (CRPS) is a proper scoring rule for ensemble probabilistic forecasts. Let $q \in \{1, \ldots, Q\}$ index all $Q = TCHW$ scalar verification points, let y_q be the denormalized ground truth, and let $\{\hat{y}_q^{(k)}\}_{k=1}^K$ be K independently generated ensemble members. The pointwise empirical CRPS is

$$c_{\text{emp},q} = \frac{1}{K}\sum_{k=1}^{K}\left|\hat{y}_q^{(k)} - y_q\right| - \frac{1}{2K^2}\sum_{k=1}^{K}\sum_{l=1}^{K}\left|\hat{y}_q^{(k)} - \hat{y}_q^{(l)}\right|, \tag{37}$$

and the reported physical-space empirical CRPS is

$$\text{CRPS}_{\text{emp}} = \frac{1}{Q}\sum_{q=1}^{Q} c_{\text{emp},q}. \tag{38}$$

For $K \geq 2$, the finite-ensemble fair CRPS replaces the biased pairwise normalization with an off-diagonal correction:

$$c_{\text{fair},q} = \frac{1}{K}\sum_{k=1}^{K}\left|\hat{y}_q^{(k)} - y_q\right| - \frac{1}{K(K-1)}\sum_{k=1}^{K-1}\sum_{l=k+1}^{K}\left|\hat{y}_q^{(k)} - \hat{y}_q^{(l)}\right|, \tag{39}$$

and its physical-space average is

$$\text{CRPS}_{\text{fair}} = \frac{1}{Q}\sum_{q=1}^{Q} c_{\text{fair},q}. \tag{40}$$

When the ensemble members are independent samples from the same predictive distribution, fair CRPS is an unbiased estimator of the CRPS of that underlying distribution, whereas empirical CRPS retains a positive finite-ensemble bias that decreases with K. When $K = 1$, the empirical pairwise term is zero and empirical CRPS equals MAE, while fair CRPS is undefined. Normalized empirical and fair CRPS are obtained by applying the corresponding formulas after mapping predictions and targets to $[0, 1]$ with the fixed normalization for each band, and are then averaged across bands. Both scores are used to evaluate stochastic ensemble forecasts rather than the calibration of a single deterministic output.

D MODEL CONFIGURATION

D.1 CONFIGURATION FOR TRAINING EVOLUTION NETWORK

As shown in Table 9, the Evolution Network is trained using the specified configurations, optimization hyperparameters, and model architectural settings.

Table 9: Configuration details for the evolution Network on Himawari-8/9 and GOES-16.

Category	**Parameter**	**Value**
Dataset	Input / Output Length	6 frames / 6 frames
	Training / Validation Ratio	0.9 / 0.1
	Data Augmentation	Random Flip (p=0.5)
Optimization	Optimizer	AdamW
	Base Learning Rate	1.0×10^{-3}
	Total / Micro Batch Size	64 / 4
	Weight Decay	1.0×10^{-5}
	Max Epochs	100
	Scheduler	Cosine Annealing
	Warmup Ratio	0.1
	Gradient Clipping	1.0
Loss Function	Primary Loss Type	L1
Model	Name	Evolution Network
	Base Channels	64
	Frame Channels	8
	Resolution ($H \times W$)	256×256
	Dynamic Channels	[32, 16, 16]
	Initial Residual Scale	0.01
	Interpolation / Padding	Bilinear / Border
Global Settings	Precision	Float32

D.2 CONFIGURATION FOR TRAINING REFINEFORMER IN PHYSICS-ANCHORED FLOW MATCHING

As shown in Table 10, the RefineFormer is trained using the specified configurations, optimization hyperparameters, and model architectural settings. During this second training stage, the pretrained Evolution Network is kept fully frozen and used only to produce the physics-inspired prior $\hat{\mathcal{F}}_{phy}$ without gradient computation. Consequently, gradients from $\mathcal{L}_{CFM}$ update only the RefineFormer and are not backpropagated into the Evolution Network.

Table 10: Configuration details for RefineFormer of CloudFlow's PA-FM on Himawari-8/9 and GOES-16 dataset.

Category	**Parameter**	**Value**
Dataset	Input / Output Length	6 frames / 6 frames
	Training / Validation Ratio	0.9 / 0.1
	Data Augmentation	Random Flip ($p = 0.5$)
Model	Resolution ($H \times W$)	256×256
	In / Out Channels	16 / 8
	Hidden Size / Depth	768 / 12
	Patch Size	16
Conditional Flow Matching	Flow Ratio	1.0
	Time Distribution	Logit-normal-based p_τ
	CFG (Ratio / Scale)	0.50 / 2.0
Optimization	Learning Rate	1.0×10^{-4}
	Weight Decay	0.0
	Warmup Steps	5000
	Gradient Clipping	1.0
Training	Total Steps	1,000,000
	Precision	Float32

E INTERPRETABILITY & DESIGN CHOICES

E.1 ANALYSIS OF INFERENCE STEPS N FOR THE PERCEPTION-ACCURACY TRADE-OFF.

Table 11 and Figure 8 quantify the perception–accuracy trade-off across inference steps N, while Figures 9, 10, and 11 illustrate the corresponding changes for the 0.64 µm, 1.6 µm, and 3.9 µm bands. With $N = 1$, CloudFlow largely retains the coarse prediction of the Evolution Network, leading to visible over-smoothing at late lead times. Increasing N to 5 or 10 progressively restores realistic cloud textures; in particular, moving from 5 to 10 steps improves FVD by 23.3% with only a 6.2% increase in MSE. Increasing N further to 50 yields only a modest FVD gain at the cost of a 12.1% MSE increase and $5\times$ more computation, whereas still larger values degrade FVD and may generate spurious high-frequency details, especially at early lead times. We therefore adopt $N = 10$ as a balanced operating point. The adaptive dopri5 solver also underperforms the fixed 10-step Euler solver, suggesting that the learned PA-FM vector field benefits from stable task-specific discretization. A horizon-aware strategy that uses fewer steps for short-term accuracy and more steps for long-term texture recovery remains a promising direction.

Table 11: Quantitative analysis of inference steps on the Himawari-8/9 dataset. The best results are **bolded**, and underlined values (N=10) indicate our final selection.

Inference Steps (N)	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
1	**45.89**	**3.590**	**24.38**	**0.700**	176.27
5	50.71	3.957	24.04	0.685	88.08
10	53.87	4.129	23.79	0.672	67.56
50	60.40	4.450	23.31	0.646	**61.57**
100	62.61	4.551	23.15	0.637	64.47
500	67.08	4.752	22.84	0.618	74.15
1000	69.05	4.838	22.72	0.610	79.59
adaptive	70.19	4.888	22.64	0.600	83.55

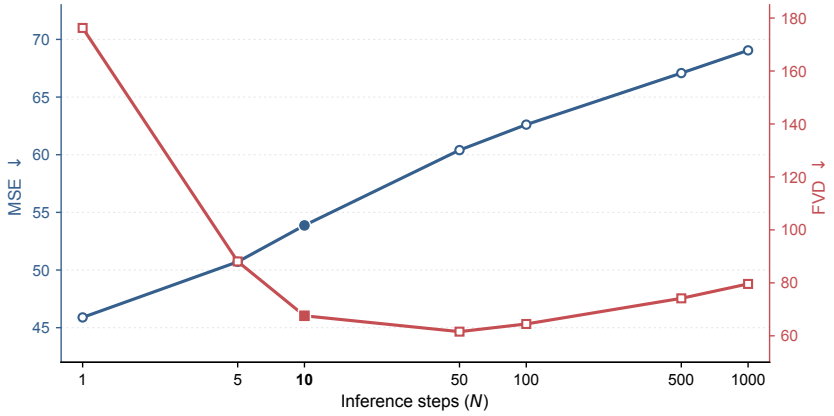


Figure 8: Perception–accuracy trade-off across inference step counts N on Himawari-8/9, measured by MSE and FVD.



Figure 9: An example of 0.64 μm visible band's prediction results of CloudFlow with different inference steps N in PA-FM.

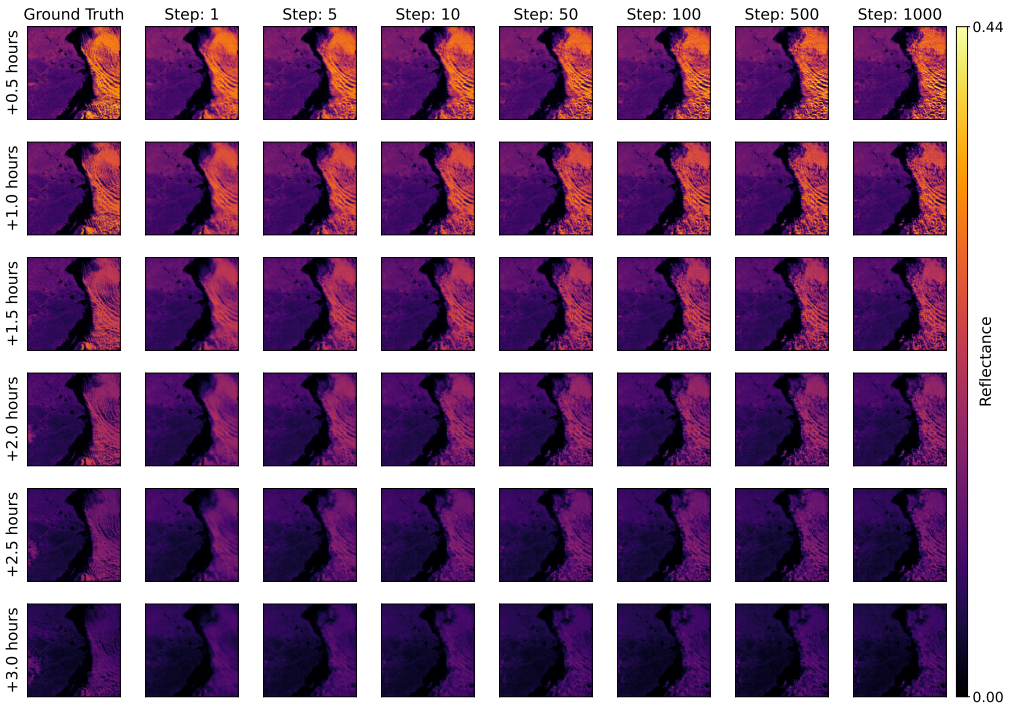


Figure 10: An example of 1.6 μm infrared band's prediction results of CloudFlow with different inference steps N in PA-FM.

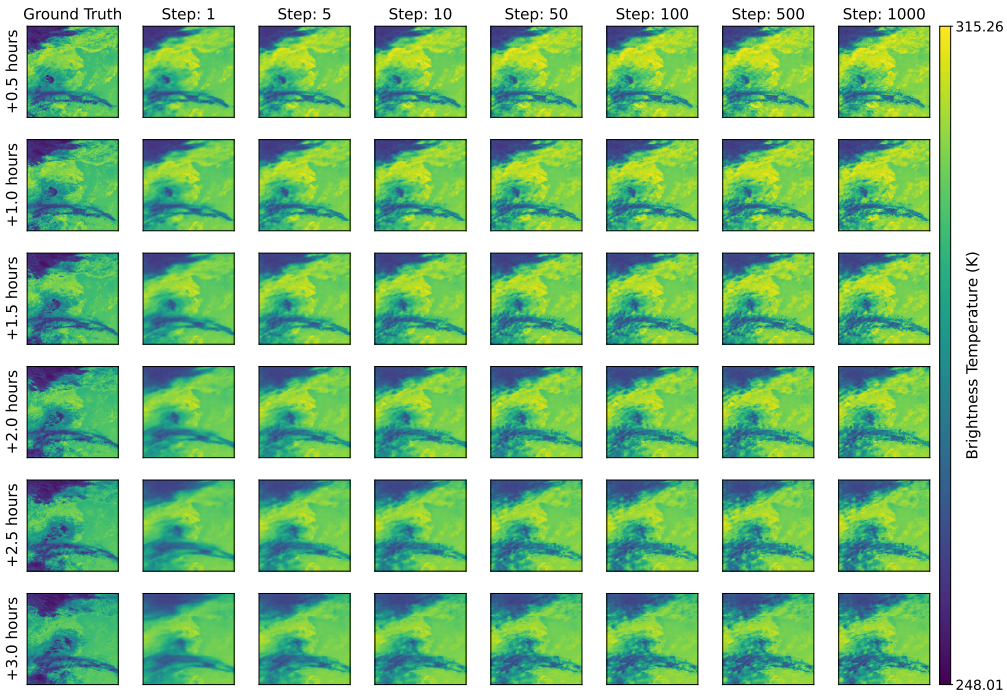


Figure 11: An example of 3.9 μm infrared band's prediction results of CloudFlow with different inference steps N in PA-FM.

E.2 Analysis of DFI fusion strategy

To further validate the proposed scale-aware Dynamics Feature Integration (DFI), we compare several fusion schemes in the Source-Sink Decoder. As shown in Table 12, different strategies lead to distinct trade-offs. The `all_mid` scheme achieves the lowest FVD, indicating stronger perceptual quality, while `reversed_stage_specific` obtains the lowest MSE and highest PSNR but suffers from a clear SSIM degradation. In contrast, our physical-intuition-guided `stage_specific` strategy achieves the best MAE and SSIM, providing the most balanced compromise between pixel-level accuracy and structural integrity. This suggests that injecting synoptic-scale features into deep layers and convective-scale features into shallow layers avoids both over-smoothing and excessive hallucination.

Table 12: Ablation study of different DFI fusion strategies. The best results are highlighted in **bold**.

Fusion Scheme	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
`none`	45.20	3.319	24.370	0.6991	117.74
`stage_specific`	44.55	**3.296**	24.420	**0.7011**	130.44
`reversed_stage_specific`	**43.76**	3.312	**24.467**	0.6817	129.52
`all_early`	45.88	3.317	24.206	0.6751	119.81
`all_mid`	45.28	3.326	24.305	0.6785	**111.20**
`all_late`	44.34	3.302	24.375	0.6801	120.51
`all_stages`	44.41	3.326	24.352	0.6793	121.60

E.3 Analysis of hyperparameter α initialization

Table 13 shows $\alpha = 0.01$ achieves the best performance on all metrics, except for FVD. This optimal value ensures the source-sink term acts as a sparse corrective signal rather than a primary driver. Larger values like $\alpha = 1$ allow term S to dominate, causing the network to bypass structured advection laws in favor of unconstrained intensity updates, which results in weak physical plausibility and degraded

accuracy and visual quality. Conversely, an undersized $\alpha = 0.0001$ fails to capture non-conservative phase changes, reducing the framework to a pure advection model incapable of simulating cloud intensification or dissipation. Thus, $\alpha = 0.01$ effectively balances the physics-inspired prior with source-sink corrections to capture complex atmospheric evolution.

Table 13: Analysis of hyperparameter α of Source-Sink Decoder on Himawari-8/9 dataset. The best results are highlighted in **bold**.

α	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
1	45.48	3.342	24.34	0.685	**120.7**
0.1	45.23	3.313	24.37	0.690	128.8
0.01	**44.55**	**3.296**	**24.42**	**0.701**	130.4
0.001	45.93	3.344	24.31	0.699	127.3
0.0001	45.23	3.320	24.36	0.686	123.6

E.4 Joint Multi-Frame Prediction versus Autoregressive Forecasting

To assess whether jointly predicting multiple future frames is preferable to iterative one-step forecasting for our three-hour nowcasting task, we conduct a controlled comparison on Himawari-8/9 using the same SimVP Gao et al. (2022a) backbone and evaluation setting. The $2 \to 1$ autoregressive (AR) variant takes two consecutive frames to predict the next frame and recursively feeds each prediction back as input until six future frames are produced. In contrast, the $6 \to 6$ joint variant takes six historical frames and predicts all six future frames in a single forward pass.

Table 14: Controlled comparison of $2 \to 1$ autoregressive forecasting and $6 \to 6$ joint prediction using the same SimVP backbone on Himawari-8/9.

Setting	**30 min**	**60 min**	**90 min**	**120 min**	**150 min**	**180 min**	**Average MSE↓**
$2 \to 1$ AR	**17.16**	**33.04**	50.45	68.51	87.17	106.72	60.51
$6 \to 6$ joint	32.42	40.82	**49.48**	**58.85**	**68.15**	**75.79**	**54.25**

The $2 \to 1$ AR setting performs better at 30 and 60 minutes, consistent with the easier objective of learning a local one-step transition. However, the $6 \to 6$ joint setting overtakes it from 90 minutes onward. At 180 minutes, its MSE is 75.79 compared with 106.72 for autoregressive forecasting, and its six-step average MSE decreases from 60.51 to 54.25, a reduction of approximately 10.3%. This demonstrates that joint multi-frame prediction provides better accuracy over the complete three-hour forecast horizon, despite sacrificing some short-lead precision.

This behavior can be explained by both temporal context and training–inference consistency. Six historical observations provide a longer baseline for estimating cloud motion and deformation. Moreover, joint multi-horizon supervision directly penalizes errors at every lead time without recycling predicted frames as inputs. In contrast, although the $2 \to 1$ model learns a simpler local transition, its autoregressive rollout conditions on imperfect predictions that are not encountered as inputs during training. This exposure bias propagates motion and intensity errors between successive steps and causes them to accumulate with lead time. Although combining these two strategies may improve prediction accuracy, exploring such a combination is not the primary focus of this work.

E.5 Interpretability of Learned Evolution

Figure 13, 14 and 15 show the learned velocity field v and intensity-change field $\mathbf{S}$ on representative Himawari-8/9 samples. Across different cloud patterns, the predicted velocity fields are spatially coherent and aligned with the observed cloud motion. In active cloud regions, the vectors follow the dominant velocity direction of the cloud structures, while in clear-sky or weak-texture regions the predicted velocities remain close to zero. Meanwhile, the learned intensity fields are mainly activated around cloud boundaries and evolving cloud bodies, indicating that the model uses $\mathbf{s}_t$ to correct appearance changes that cannot be explained by pure advection. The close correspondence between

the predictions and ground truth further suggests that the model does not simply memorize future appearances, but learns an interpretable evolution process consisting of motion-induced transport and intensity-induced transformation.

A natural concern is that the velocity field is not explicitly constrained by divergence-free, smoothness, or other hand-crafted regularization terms. However, the Motion Decoder is built upon convolutional receptive fields, which introduce an implicit structural bias toward locally smooth and spatially correlated motion. More importantly, we examine whether the learned apparent cloud-motion magnitudes are physically plausible rather than numerical artifacts. As shown in Figure 12, we statistically analyze per-grid-point velocity magnitudes over approximately 2.7 billion samples on the Himawari-8/9 dataset. We use $0.56\,\mathrm{m/s}$, equivalent to approximately $1\,\mathrm{km}$ of apparent displacement over a 30-minute interval, as an operational threshold for quiescent regions rather than as a universal meteorological criterion. Under this definition, 36.5% of the predicted velocities fall below the threshold. For the remaining active regions, the mean speed is $9.37\,\mathrm{m/s}$, corresponding to a displacement of about 4.22 grid cells at $4\,\mathrm{km}$ resolution within 30 minutes. The maximum speed is $64.8\,\mathrm{m/s}$. For context, Himawari-8 atmospheric motion vectors have been shown to be broadly consistent with JMA mesoscale analyses, radiosondes, and wind-profiler observations Otsuka et al. (2018), while a multi-center Himawari-8 AMV intercomparison reported maximum speeds of 70–97 $\mathrm{m/s}$ across retrieval systems Santek et al. (2019). Thus, the upper tail of our learned field does not by itself indicate numerical divergence. We emphasize, however, that $\mathbf{v}$ is a two-dimensional apparent cloud-motion field learned for radiance transport, without the cloud-height assignment used in operational AMV retrieval; it should therefore not be interpreted as a direct measurement of the three-dimensional atmospheric wind. These qualitative and statistical results support a coherent and physically plausible evolution representation, while collocated validation against retrieved or in situ winds remains future work.

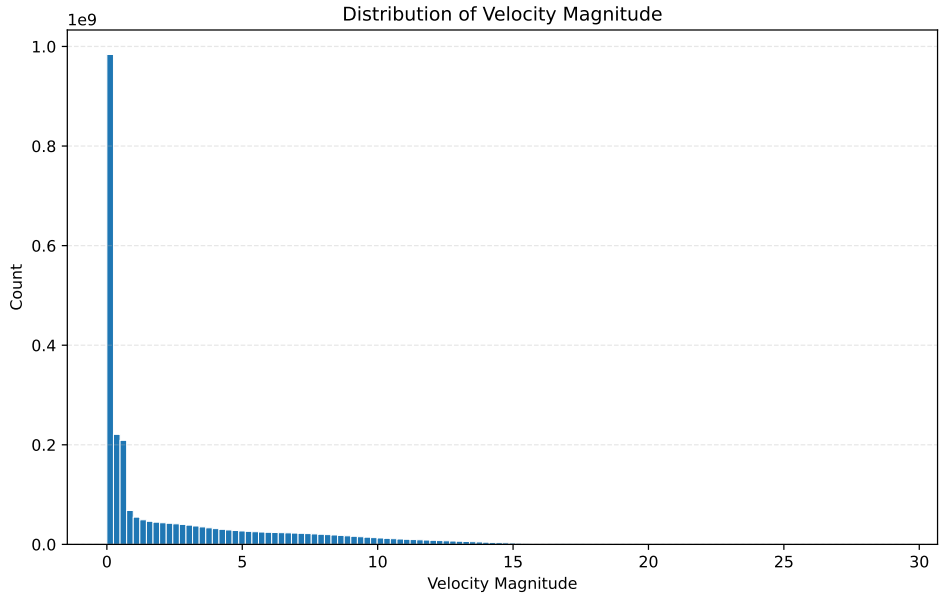


Figure 12: Distribution of the learned apparent cloud-motion magnitudes across the Himawari-8/9 dataset. The $0.56\,\mathrm{m/s}$ cutoff is an operational diagnostic corresponding to $1\,\mathrm{km}$ of apparent displacement over 30 minutes, rather than a universal meteorological threshold. The learned field does not include the cloud-height assignment required for direct atmospheric-wind retrieval.

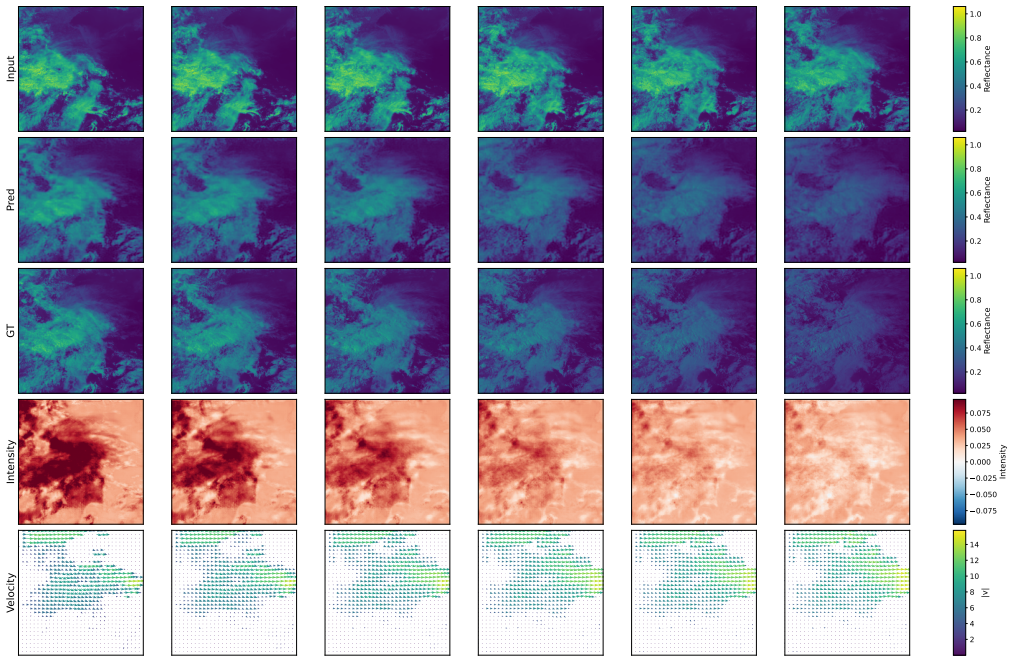


Figure 13: An example of CloudFlow prediction on the 0.64 µm visible band, showing the input, prediction, ground truth, learned intensity-change field, and velocity field.

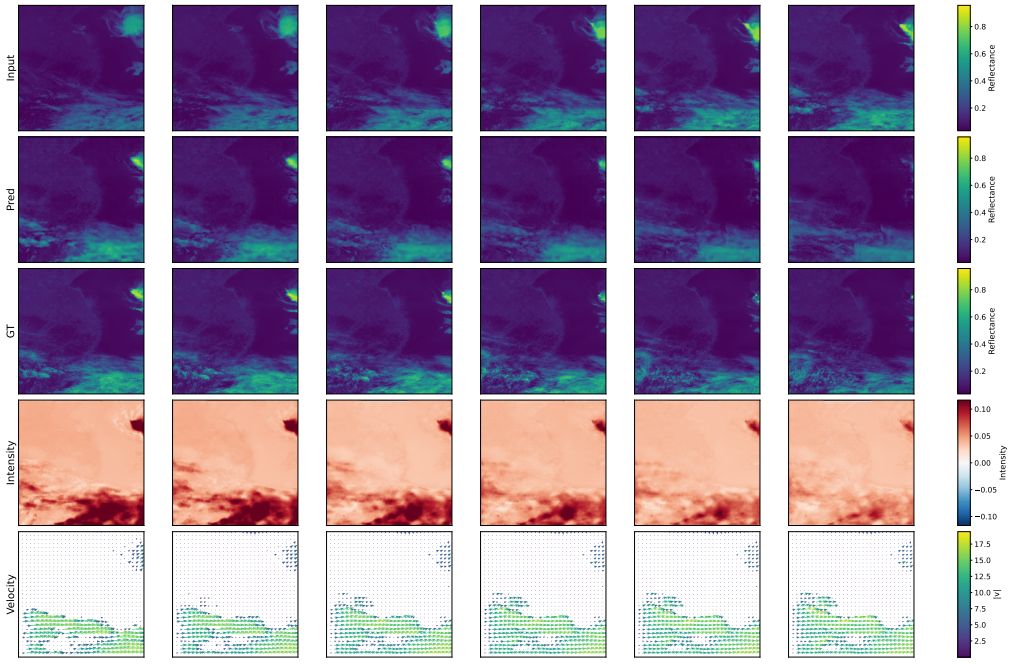


Figure 14: An example of CloudFlow prediction on the 0.64 µm visible band, showing the input, prediction, ground truth, learned intensity-change field, and velocity field.

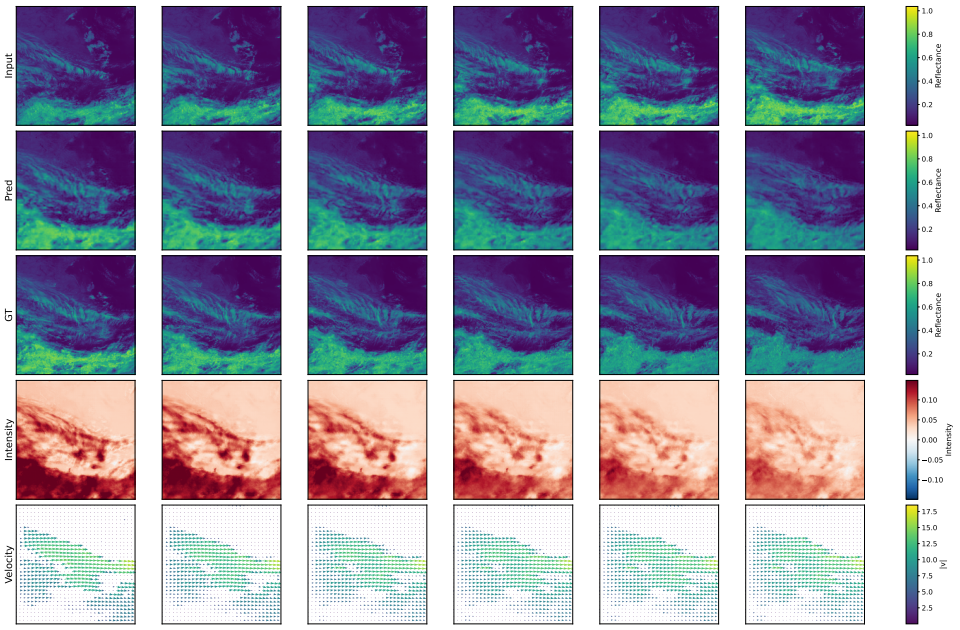


Figure 15: An example of CloudFlow prediction on the 0.64 µm visible band, showing the input, prediction, ground truth, learned intensity-change field, and velocity field.

F Performance & Efficiency

F.1 Channel- and Lead-Time-wise Accuracy against Deterministic Models

Table 1 reports metrics averaged over all spectral channels and forecast lead times. To examine whether the aggregate accuracy gain is consistent across the prediction target, Figure 16 compares the per-channel MAE of CloudFlow and the deterministic baselines at six 30-minute lead times from $+0.5$ to $+3.0$ hours on Himawari-8/9. MAE for the 0.64 µm and 1.6 µm reflectance bands is unitless, whereas the infrared bands are evaluated in brightness temperature (K).

Across most spectral bands and forecast horizons, CloudFlow lies on or near the lower error envelope, indicating consistently competitive channel-wise accuracy. The most pronounced separation appears in the 3.9 µm shortwave-infrared band, where CloudFlow yields markedly lower MAE than most baselines throughout the three-hour horizon, especially at shorter lead times. Because this band is sensitive to low-level clouds, fog, and shortwave-infrared radiometric contrasts, the result suggests that CloudFlow more faithfully preserves the corresponding cloud structures and intensity evolution. Together with its competitive performance in the remaining reflectance and thermal-infrared bands, this channel- and lead-time-resolved comparison confirms that the aggregate accuracy reported in Table 1 is not driven by a single spectral band or forecast step.

F.2 Probabilistic Evaluation and Ensemble-Size Convergence of Generative Models

We evaluate DGMR, MCVD, and PreDiff on the complete test sets using K independently seeded ensemble members for each input. Tables 15 and 16 summarize the results for $K \in \{1, 3, 5, 10\}$, while Figure 17 shows the complete empirical-CRPS trajectories for $K = 1, \ldots, 10$. For each K, MSE, MAE, PSNR, SSIM, and FVD are averaged across the K members, where FVD denotes the mean per-member FVD. Empirical and fair CRPS are computed jointly from the K-member ensemble in the physical data space. To ensure a consistent comparison across models and datasets, all formal probabilistic comparisons use $K = 10$.

The member-mean MSE, MAE, PSNR, SSIM, and FVD results stabilize by $K = 3$ on both datasets. As further illustrated in Figure 17, the largest empirical-CRPS reductions occur at small ensemble

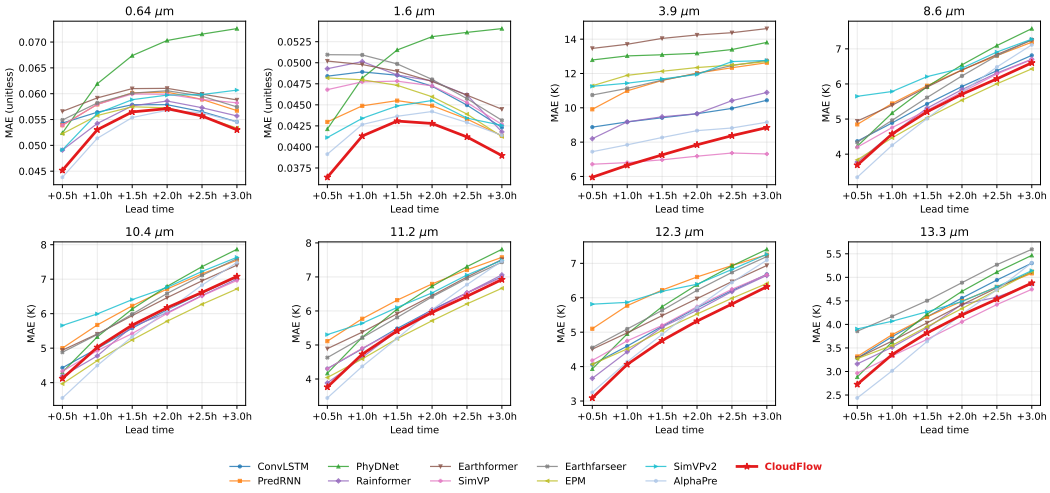


Figure 16: Per-channel MAE across forecast lead times on the Himawari-8/9 dataset for CloudFlow and deterministic baselines. The panels cover the 0.64 μm and 1.6 μm reflectance bands (unitless MAE) and six infrared bands from 3.9 μm to 13.3 μm (MAE in K). Results are reported at six 30-minute lead times through three hours. Lower values indicate better accuracy.

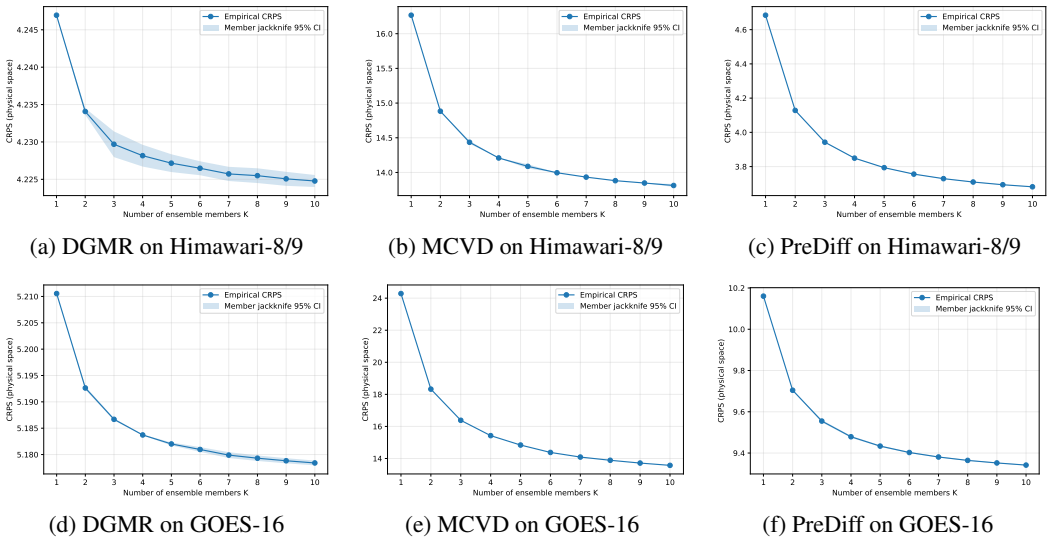
Table 15: Probabilistic evaluation and ensemble-size convergence on Himawari-8/9. Empirical and fair CRPS are evaluated in physical space; – indicates that fair CRPS is undefined for $K = 1$. The bold $K = 10$ setting is used for formal probabilistic comparison.

Model	K	MSE↓	MAE↓	PSNR↑	SSIM↑	FVD↓	Empirical CRPS↓	Fair CRPS↓
DGMR	1	58.90	4.247	23.42	0.650	509.18	4.247	–
	3	58.90	4.247	23.42	0.650	509.04	4.230	4.221
	5	58.90	4.247	23.42	0.650	508.99	4.227	4.222
	10	58.91	4.247	23.42	0.650	508.97	4.225	4.222
MCVD	1	498.31	16.266	12.74	0.425	736.11	16.266	–
	3	496.73	16.237	12.76	0.425	734.62	14.435	13.534
	5	497.34	16.249	12.75	0.425	734.85	14.088	13.548
	10	496.64	16.240	12.76	0.425	734.70	13.812	13.543
PreDiff	1	74.44	4.684	22.49	0.646	115.99	4.684	–
	3	74.27	4.684	22.50	0.646	116.18	3.942	3.572
	5	74.26	4.683	22.50	0.646	116.17	3.794	3.571
	10	74.28	4.683	22.50	0.646	116.42	3.682	3.571

sizes, with the curves approaching a plateau by $K = 10$. In contrast, the fair CRPS values in Tables 15 and 16 are already nearly invariant from $K = 3$ onward for every model–dataset pair, indicating that much of the remaining empirical-CRPS decrease is attributable to finite-ensemble bias. Using the complete $K = 1, \ldots, 10$ sequences and defining empirical-CRPS convergence as two consecutive relative changes no greater than 1%, empirical CRPS stabilizes at $K = 3$, 6, and 7 for DGMR, MCVD, and PreDiff, respectively, on Himawari-8/9. On GOES-16, it stabilizes at $K = 3$ for DGMR and $K = 5$ for PreDiff, whereas MCVD continues to decrease through $K = 10$ but with progressively smaller marginal improvements. Thus, $K = 10$ provides a conservative and uniform setting in which both empirical and fair CRPS are stable or near stable across all evaluated models and datasets.

Table 16: Probabilistic evaluation and ensemble-size convergence on GOES-16. Empirical and fair CRPS are evaluated in physical space; – indicates that fair CRPS is undefined for $K = 1$. The bold $K = 10$ setting is used for formal probabilistic comparison.

Model	K	MSE↓	MAE↓	PSNR↑	SSIM↑	FVD↓	Empirical CRPS↓	Fair CRPS↓
DGMR	1	80.63	5.211	24.49	0.749	314.22	5.211	–
	3	80.64	5.211	24.49	0.749	314.36	5.187	5.175
	5	80.64	5.211	24.49	0.749	314.47	5.182	5.175
	10	80.64	5.211	24.49	0.749	314.48	5.178	5.175
MCVD	1	1270.87	24.287	12.15	0.459	534.61	24.287	–
	3	1277.20	24.338	12.13	0.457	534.16	16.382	12.404
	5	1278.42	24.369	12.13	0.457	536.18	14.839	12.456
	10	1274.88	24.335	12.13	0.457	534.74	13.572	12.376
PreDiff	1	227.25	10.160	20.52	0.696	146.01	10.160	–
	3	227.41	10.164	20.51	0.696	145.77	9.555	9.250
	5	227.36	10.163	20.51	0.696	145.70	9.434	9.251
	10	227.29	10.162	20.51	0.696	145.54	9.342	9.250



(a) DGMR on Himawari-8/9 (b) MCVD on Himawari-8/9 (c) PreDiff on Himawari-8/9

(d) DGMR on GOES-16 (e) MCVD on GOES-16 (f) PreDiff on GOES-16

Figure 17: Empirical CRPS as a function of ensemble size K for the three stochastic generative baselines on Himawari-8/9 and GOES-16. Shaded regions denote member-jackknife 95% confidence intervals. Across model–dataset pairs, CRPS decreases most rapidly for small ensembles and becomes substantially flatter as K approaches 10, supporting the use of $K = 10$ for stable and uniform probabilistic evaluation.

F.3 Additional visualizations of nowcast results

As shown in Figures 18, 19, 20 and 21, the prediction results exhibit distinct characteristics across different methods. Deterministic models can maintain the main structure of clouds but tend to generate blurry predictions. Although other generative baselines improve sharpness, they introduce noticeable artifacts. Specifically, MCVD suffers from severe mean shift manifested as sudden intensity change, while PreDiff struggles to maintain the structural coherence of clouds, often resulting in fragmented patterns. In contrast, CloudFlow produces clear and sharp predictions with rich fine-grained details. It effectively preserves cloud structures without fragmentation and avoids the mean shift phenomenon, thereby visually aligning most closely with the ground truth.

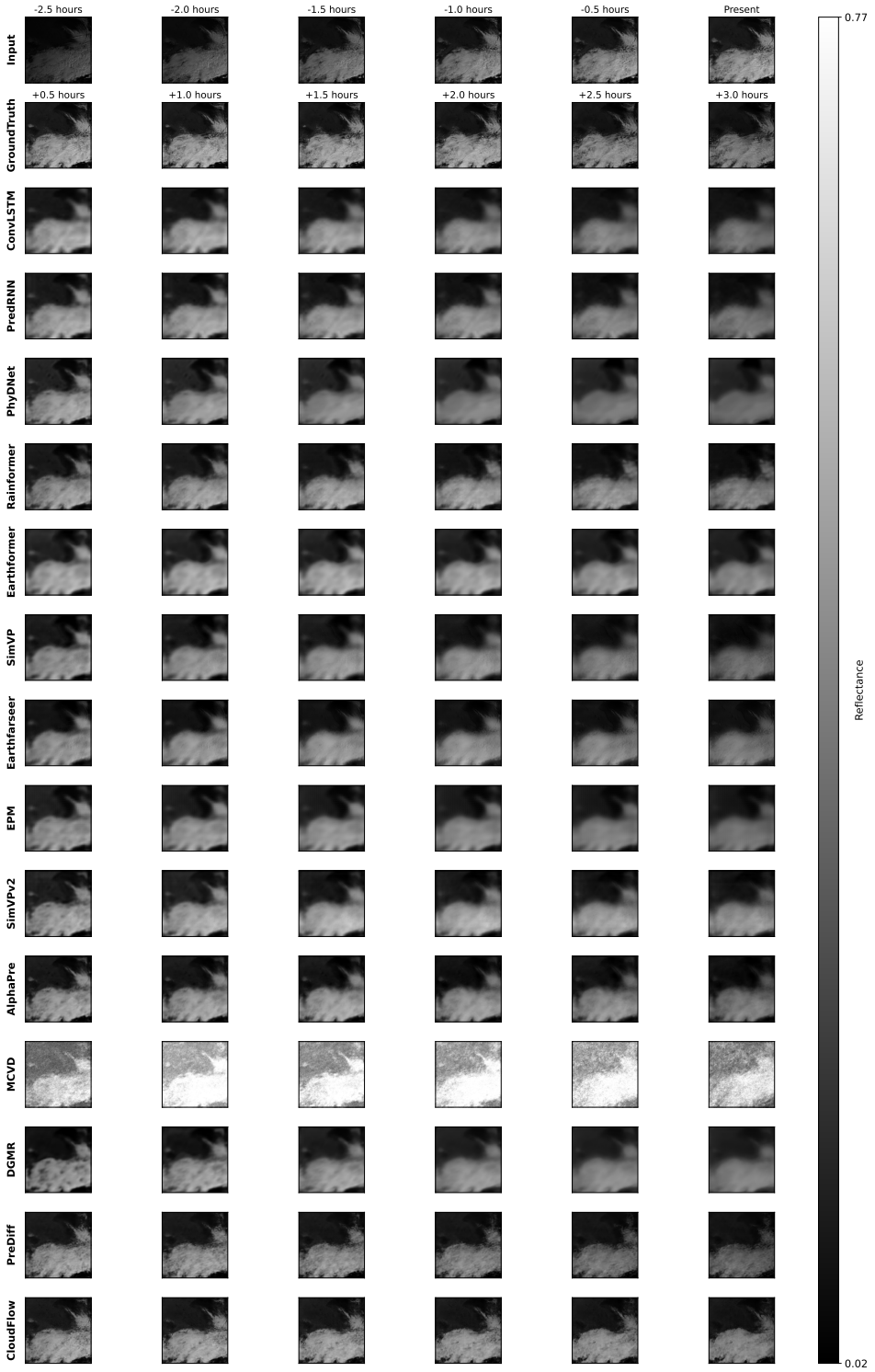


Figure 18: 0.64 μm visible band's prediction results of CloudFlow and other models on Southeast Asian coast, 01/03/2023 02:50 (UTC).

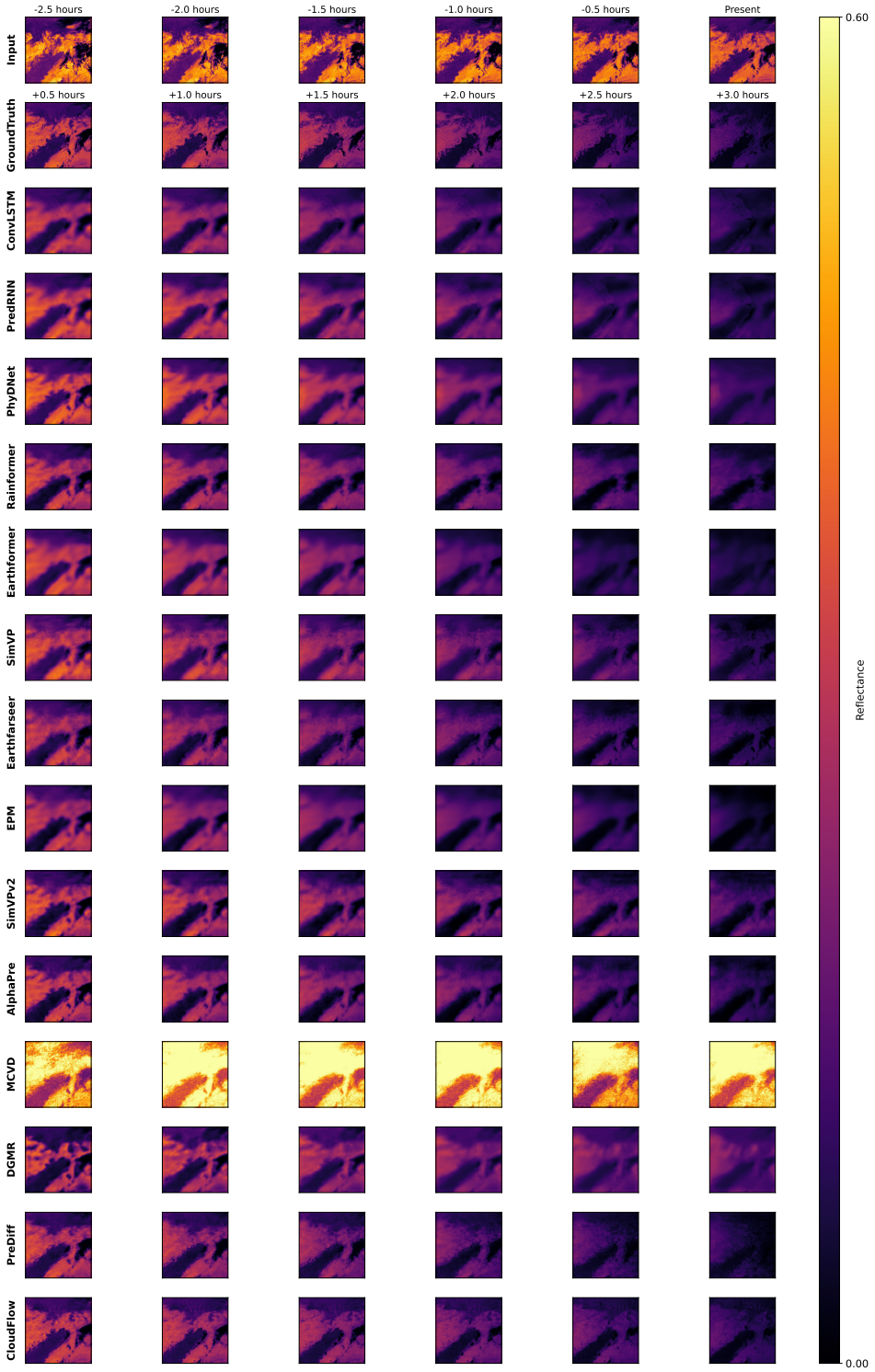


Figure 19: 1.6 μm near-infrared band's prediction results of CloudFlow and other models on Southeast Asian coast, 02/22/2023 05:50 (UTC).

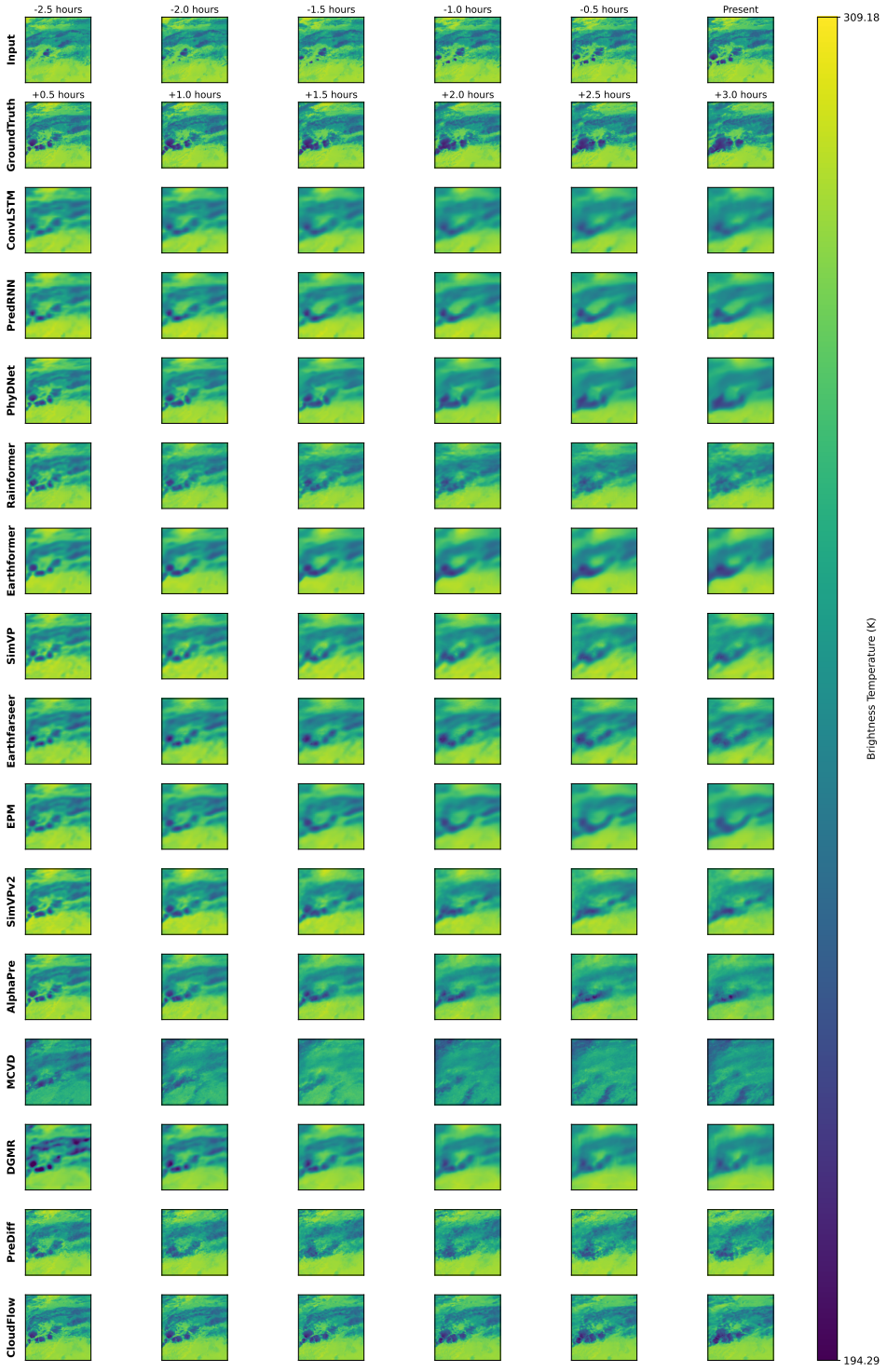


Figure 20: 10.4 μm infrared band's prediction results of CloudFlow and other models on Southeast Asian coast, 06/05/2023 05:20 (UTC).

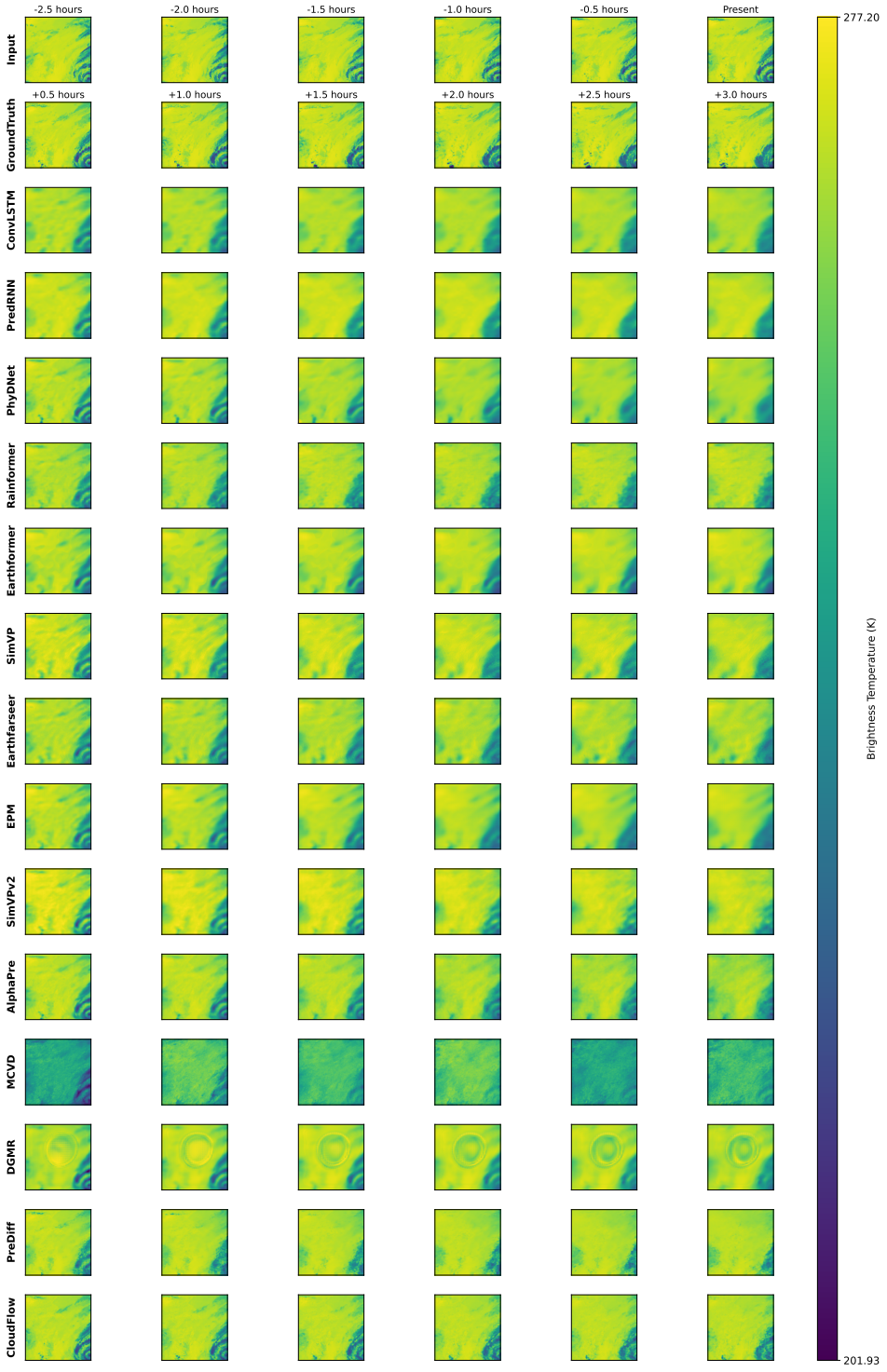


Figure 21: 13.3 μm infrared band's prediction results of CloudFlow and other models for Typhoon Mawar on Southeast Asian coast, 06/01/2023 05:40 (UTC).

F.4 Comparison of Model Efficiency

Table 17: Comparison of Parameters, Sample Steps, and GFLOPs across different models. N denotes the number of sampling steps.

Category	Model	Params (M)	Sample Steps (N)	1 step GFLOPs	Total GFLOPs
Deterministic	ConvLSTM	8.36	1	94.05	94.05
	PredRNN	13.82	1	134.61	134.61
	PhyDNet	7.22	1	690.96	690.96
	Rainformer	139.42	1	114.64	114.64
	Earthformer	8.64	1	100.04	100.04
	SimVP	14.88	1	241.06	241.06
	Earthfarseer	97.66	1	8778.85	8778.85
	EPM	11.84	1	310.85	310.85
	SimVPv2	13.18	1	196.49	196.49
	AlphaPre	34.05	1	5434.33	5434.33
Generative	MCVD	26.32	1000	283.91	283,910
	DGMR	3.63	1	6.67	6.67
	PreDiff	91.56 + 136.79	1000	3963.28 + 602.14	606,103.28
Ours	CloudFlow (Himawari-8/9)	47.46 + 146.24	**10**	228.7 + **51.75**	**746.20**
	CloudFlow (GOES-16)	47.46 + 146.24	**4**	228.7 + **51.75**	**435.70**

Table 17 provides a comprehensive comparison of computational costs. For the generative models, we decompose the GFLOPs into two components A+B. For PreDiff, this refers to the Encoder/Decoder and the latent diffusion steps. For our CloudFlow, it denotes the Evolution Network and the generative refinement steps. Compared to diffusion-based models like MCVD and PreDiff, CloudFlow achieves orders-of-magnitude acceleration. Specifically, CloudFlow reduces the total GFLOPs by approximately 812x relative to PreDiff. This efficiency stems from both our lightweight single-step cost and the significantly reduced sampling iterations. Remarkably, CloudFlow brings the computational overhead of generative forecasting down to the same order of magnitude as deterministic models. For instance, the total GFLOPs of CloudFlow for GOES is only 435.70, which is even lower than several deterministic baselines such as AlphaPre and Earthfarseer.

G GENERALIZATION & ROBUSTNESS

G.1 CROSS-REGION GENERALIZATION IN NORTHWESTERN CHINA

To evaluate cross-region generalization, we directly evaluate the checkpoints trained on the Himawari-8/9 dataset used in the main experiments on the cross-region test set, without retraining or fine-tuning.

Table 18: Cross-region evaluation on a geographically distinct Himawari-8/9 test region with different terrain and climate conditions. All models use checkpoints trained on the original Himawari-8/9 region without retraining or fine-tuning. Best results are highlighted in **bold**, and second-best results are underlined.

Model	MSE↓	MAE↓	PSNR↑	SSIM↑	FVD↓
Deterministic Models					
ConvLSTM	82.01	5.568	22.29	0.596	426.5
PredRNN	105.76	6.763	21.36	0.597	440.6
PhyDNet	111.94	6.846	20.84	0.581	409.7
Rainformer	77.62	5.283	22.52	0.575	281.4
Earthformer	128.07	7.609	20.61	0.601	517.3
SimVP	81.58	5.635	22.47	0.586	279.0
Earthfarseer	93.34	6.203	21.83	0.579	285.6
EPM	81.65	5.586	22.39	0.589	396.8
SimVPv2	83.38	5.630	22.31	0.609	341.1
AlphaPre	78.49	5.296	22.59	0.620	225.0
Generative Models					
MCVD	504.96	16.11	12.96	0.395	616.9
DGMR	78.23	5.190	**22.69**	0.599	434.8
PreDiff	106.48	6.122	21.98	0.524	193.8
Ours					
CloudFlow	**75.47**	**4.880**	22.44	**0.622**	**77.06**

Without adaptation to the new region, CloudFlow achieves the best MSE, MAE, SSIM, and FVD among the evaluated methods, while maintaining competitive PSNR. In particular, its FVD of 77.06 is approximately 60.2% lower than the second-best value of 193.8 from PreDiff. These results provide preliminary evidence that the learned evolution prior and conditional refinement transfer across geographically distinct regions while preserving both pixel-level accuracy and perceptual quality. Since this experiment remains within the coverage of Himawari-8/9, it should not be interpreted as evidence of universal geographic generalization.

G.2 ANALYSIS OF PERFORMANCE UNDER DIFFERENT SEASONS

To evaluate the performance of CloudFlow across varying seasonal conditions, we conducted analysis on a seasonally stratified dataset. It is important to note that the dashed line in Figure 22 represents the arithmetic mean of the four seasonal metrics, rather than the performance evaluated on the complete dataset. This distinction is particularly significant for FVD; as indicated in Eq. (36), the metric is sensitive to sample size, causing the arithmetic mean of the subsets to diverge notably from the global indicator due to the reduced sample volume in each season.

As shown in Figure 22, CloudFlow exhibits a distinct performance decline during the summer months. This degradation is characterized by elevated prediction errors measured by MSE and MAE, reduced visual quality in terms of PSNR and SSIM, and compromised perceptual quality indicated by FVD. We primarily attribute this drop in performance to the predominance of rapidly evolving convective systems in summer which introduce high stochasticity and complex non-linear dynamics. In contrast, the superior performance observed across spring, autumn, and winter suggests that CloudFlow is more robust when handling scenes governed by stable and large-scale advective regimes where atmospheric processes are less volatile compared to summer convection.

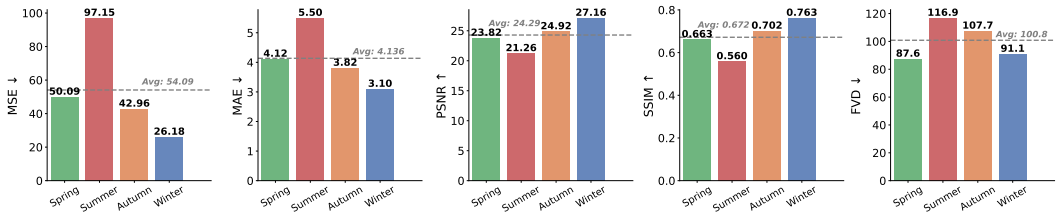


Figure 22: Seasonal analysis of CloudFlow on the Himawari-8/9 dataset. The performance degradation in Summer stems from complex convective activities and rapid cloud evolution, whereas the relatively stable atmospheric conditions dominated by advection in Winter lead to superior predictive accuracy across all metrics.

G.3 Analysis of robustness under noise condition

To evaluate the robustness of CloudFlow under realistic noisy conditions, we inject Gaussian noise to simulate the imperfections typical of satellite imagers, after the input data have been normalized to the $[0, 1]$ range via min-max scaling. Given an input tensor $\mathbf{X} \in \mathbb{R}^{B\times T\times C\times H\times W}$, noise is added as follows:

$$\tilde{\mathbf{X}} = \text{clamp}(\mathbf{X} + \boldsymbol{\epsilon},\ 0,\ 1), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2) \tag{41}$$

for additive noise and

$$\tilde{\mathbf{X}} = \text{clamp}(\mathbf{X} \cdot (1 + \boldsymbol{\epsilon}),\ 0,\ 1), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2) \tag{42}$$

for multiplicative noise, where σ denotes the noise standard deviation and the *clamp* function ensures the perturbed inputs remain within the valid $[0, 1]$ range.

The experimental results presented in Table 19 demonstrate that **CloudFlow maintains high performance stability under both noise types**. Across all evaluation metrics, including pixel-level accuracy and perceptual quality, the model exhibits a controlled and gradual degradation as noise intensity increases. Notably, **CloudFlow proves to be even less sensitive to multiplicative noise than to additive perturbations**. For instance, at the highest noise level of $\sigma = 0.1$, the MSE under multiplicative noise is 62.77, which is significantly lower than the 75.37 observed under additive noise. Similarly, the FVD remains much lower at 136.88 compared to 216.93 in the additive case, suggesting that the model's spatiotemporal modeling effectively accommodates scaling variations.

Table 19: CloudFlow's robustness analysis under different types and standard deviations of Gaussian noise on Himawari-8/9 dataset.

Noise Type	Noise Std (σ)	MSE ↓	MAE ↓	PSNR ↑	SSIM ↑	FVD ↓
-	-	53.87	4.129	23.79	0.672	67.56
Additive	0.01	53.92	4.136	23.79	0.671	66.17
Additive	0.05	59.19	4.435	23.35	0.621	107.8
Additive	0.10	75.37	5.169	22.27	0.507	216.9
Multiplicative	0.01	53.83	4.130	23.80	0.672	66.47
Multiplicative	0.05	55.83	4.254	23.64	0.650	75.39
Multiplicative	0.10	62.77	4.588	23.17	0.592	136.8

Interestingly, at a low noise intensity of $\sigma = 0.01$, certain metrics such as MSE and FVD exhibit slight improvements compared to the clean baseline. This phenomenon can be attributed to the two-stage architecture of CloudFlow, specifically the Stage (II) Physics-Anchored Flow Matching (PA-FM) component. As a continuous-time iterative generative model, PA-FM benefits from the minor perturbations which introduce a degree of sample diversity at the initial state. **During the subsequent iterative refinement process, the PA-FM effectively leverages its learned physics-inspired priors to rectify these perturbations, mapping the noisy inputs back onto the constrained manifold of atmospheric dynamics.** Consequently, PA-FM not only demonstrates high tolerance to initial

uncertainty but also yields superior structural consistency and perceptual quality, further confirming the robustness and adaptive refinement capability of the CloudFlow framework.

G.4 Analysis of robustness under data missing condition

To evaluate the model's resilience to observation loss, we simulate sensor failure by filling specific input frames with zeros, a strategy referred to as **zero-fill**. As detailed in Table 20, CloudFlow demonstrates commendable structural robustness across various data-loss patterns. When non-critical historical frames from X_{t-5} to X_{t-2} are missing, the performance degradation remains within a remarkably stable range. For instance, even with the loss of the relatively recent X_{t-1} frame, the MSE only increases to 156.57, maintaining a functional prediction capacity within the same order of magnitude as the baseline.

However, a critical structural sensitivity is observed when the latest observation frame X_t is missing, causing the MSE to spike catastrophically to 4267.83. This vulnerability is primarily rooted in the internal mechanism of the Evolution Network. Specifically, when the advection operator performs its first warping operation, it attempts to sample features from the previous time step along the velocity field trajectories. Since X_t is a zero-tensor, this sampling process yields null values. Although the intensity fields provide residual information, their magnitude after scaling remains insufficient to compensate for the missing data, resulting in a coarse-grained prediction $\hat{\mathcal{F}}_{phy}$ that is nearly empty. Furthermore, during the inference process of PA-FM, the initial samples are conditioned on the output of the first stage. When $\hat{\mathcal{F}}_{phy}$ approaches a zero-tensor, the starting point for the generative flow lacks the necessary physical anchors, leading to a collapse of the refined results.

Interestingly, an unconventional phenomenon emerges where the FVD actually improves when distant frames such as X_{t-5}, X_{t-4}, or X_{t-3} are absent. We attribute this to the reduction of temporal noise and redundancy. By zero-filling these frames, the model is forced to concentrate on the most recent temporal signals, potentially leading to smoother motion trajectories. However, it is crucial to note that this isolated FVD improvement does not signify superior performance. On the contrary, the concurrent increase in MSE and the noticeable degradation in visual fine-grained details suggest that the model's predictive accuracy is compromised. This indicates that while the generated sequences may appear more temporally consistent in the feature space, they sacrifice the critical high-frequency information and structural fidelity required for precise meteorological forecasting.

Table 20: CloudFlow's robustness analysis under missing frame settings on Himawari-8/9 dataset.

Missing Frame	**Strategy**	**MSE** ↓	**MAE** ↓	**PSNR** ↑	**SSIM** ↑	**FVD** ↓
None	–	53.87	4.129	23.791	0.672	67.56
X_{t-5}	Zero-fill	62.79	4.709	22.42	0.643	57.91
X_{t-4}	Zero-fill	60.11	4.318	23.23	0.659	49.65
X_{t-3}	Zero-fill	72.88	4.735	22.52	0.651	60.52
X_{t-2}	Zero-fill	92.67	5.689	21.42	0.605	105.5
X_{t-1}	Zero-fill	156.5	7.538	19.40	0.563	165.8
X_t (Latest)	Zero-fill	4267	54.68	5.618	0.174	845.7
	Remedy 1: Copy Frame	93.63	5.391	21.49	0.624	213.4
	Remedy 2: Optical Flow	70.19	4.721	22.59	0.646	106.8

H LIMITATIONS AND FUTURE WORK

This work focuses on short-term multispectral satellite image nowcasting using only geostationary meteorological satellite observations. The framework models the apparent two-dimensional evolution of cloud systems through a physics-inspired transport-residual decomposition, without explicitly reconstructing three-dimensional atmospheric states or incorporating external numerical weather prediction fields. This design enables low-latency deployment and direct image-level prediction, but it inevitably simplifies the full atmospheric evolution process, where vertical motion, thermodynamic conditions, and microphysical phase transitions may influence the predictions.

Future work will integrate low-latency thermodynamic and microphysical observations into a hybrid physical-neural source-sink module, enabling a more comprehensive characterization of cloud development, dissipation, and phase transitions. We also plan to extend the framework from regional short-term nowcasting to long-term global nowcasting, where multi-scale atmospheric circulation, cross-region generalization, and temporal error accumulation become more prominent challenges.