DC-Ocean: Deep Latent Compression for Global High-Resolution Ocean Forecasting

Yuxiang Li[1,2,3*] **Qiusheng Huang**[1,2,3*] **Hao Li**[1,2,3†]
[1]Artificial Intelligence Innovation and Incubation Institute, Fudan University
[2]Shanghai Innovation Institute
[3]Shanghai Academy of AI for Science
[*]Equal contribution [†]Corresponding author

Abstract

High-resolution global ocean states expose a bottleneck in data-driven forecasting: effective long-horizon rollouts require a compact, decodable latent state that reduces I/O and stabilizes error growth. While learned compression has matured for natural RGB images, its assumptions do not transfer to ocean modeling: ocean states are multi-variable and multi-depth, their grids are orders of magnitude larger, and fixed land–sea boundaries create sharp discontinuities that make aggressive downsampling error-prone. We present DC-Ocean, the first deep latent compression architecture for multi-variable, multi-depth global $1/12^\circ$ ocean states, built on an autoencoder that couples convolutional feature extraction with transformer-based global context modeling to retain both coastal sharpness and basin-scale structure. To explicitly address coastline-induced artifacts, DC-Ocean integrates boundary-aware gated convolutions together with a spherical geodesic front propagation interpolation scheme for stable behavior near land masks. Across key ocean variables, DC-Ocean achieves higher reconstruction fidelity than strong autoencoder baselines adapted from the natural-image domain at matched downsampling factors. Beyond reconstruction, we perform autoregressive forecasting directly in latent space and decode predictions back to the physical domain. DC-Ocean matches short-term accuracy of full-resolution forecasting while achieving lower RMSE and slower error growth at medium and long lead times. Together, these results demonstrate that DC-Ocean provides a practical framework for stable and extended high-resolution sub-daily ocean forecasting.

1 Introduction

AI-based global ocean modeling is moving toward eddy-resolving regimes and sub-daily update cycles. This shift is scientifically necessary for resolving mesoscale variability and coastal processes, but it makes data volume a first-order systems constraint. In classical numerical prediction systems[5, 3, 23, 35, 4, 33, 16, 14, 13, 20], pushing resolution to the eddy-resolving regime comes with prohibitive computational and operational cost, which further amplifies the need for more efficient data representations. A multi-year, multi-variable global ocean dataset at $1/12^\circ$ resolution can easily reach hundreds of terabytes to multiple petabytes, and storage together with I/O increasingly dominates the end-to-end cost of machine-learning workflows [23]. In practice, the bottleneck is often not GPU throughput but data movement, which slows iteration and limits how far one can push high-resolution modeling.

Compression is a natural response. The Earth-science community has long relied on error-bounded compressors such as ZFP to reduce file size for archival and distribution [24]. However, such value-oriented compression primarily targets cold storage and typically requires full-field decoding before

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

learning, so it does not substantially reduce the training-time cost that is dominated by repeated data loading and staging. Our focus is therefore on *representation* compression, where the goal is to learn a compact latent space that preserves reusable physically meaningful spatial structures and can directly support downstream modeling. Learned compression in natural images provides a clear template for this direction. Neural codecs based on autoencoders, variational objectives, and learned entropy models have achieved strong rate–distortion performance through end-to-end optimization [1, 2, 29]. The variational autoencoder (VAE) formalizes latent bottlenecks through amortized inference and KL-regularized training, and it has become a standard tool for learning structured latent representations [22]. However, the ocean setting we target introduces structural challenges that prevent direct transfer.

There are three key mismatches. First, ocean states are multi-variable and multi-depth stacks, and each variable and depth band has distinct statistics and error sensitivities. Second, global eddy-resolving grids are so large that small reductions are often insufficient to change the storage and I/O regime. Third, the land–sea interface is a fixed, sharp, and spatially discontinuous boundary. Errors near coastlines do not behave like benign blur, and they can dominate artifacts exactly where gradients and dynamics are strongest. In this work, we propose DC-Ocean, a deep latent compression framework designed for global high-resolution ocean states. DC-Ocean uses an autoencoder backbone that couples convolutional modeling with transformer-style global context learning [27]. To explicitly mitigate coastline-induced errors, it integrates boundary-aware gated convolutions and a spherical geodesic front propagation interpolation scheme inspired by front-propagation solvers [34]. We evaluate reconstruction quality in physical space and assess whether the latent space is semantically useful by training forecasters directly on compressed latents and decoding outputs back to the original grid. Our contributions are threefold.

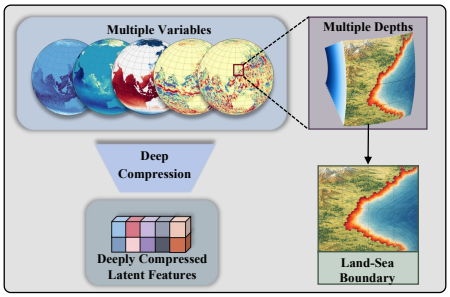

Figure 1: Key challenges in compressing high-resolution global ocean data include heterogeneous multi-variable and multi-depth ocean states, aggressive spatial compression, and sharp land–sea boundary discontinuities sensitive to downsampling.

- We develop the first deep latent compression architecture for multi-variable, multi-depth global ocean states, enabling aggressive spatial–depth compression while retaining high-fidelity reconstructions.
- We propose boundary-aware gated convolution and spherical geodesic front propagation interpolation to mitigate systematic artifacts near land–sea boundaries.
- Through experiments, we show that forecasting directly on our compressed latents enables substantially longer-horizon rollouts than operating on full-resolution high-resolution tensors, indicating that the learned latents contain reorganized, dynamically meaningful relevant physical structures rather than merely compressed values.

2 Related Work

2.1 Learned compression for natural images and beyond

Learned image compression has largely followed an autoencoder-style transform coding pipeline, where an encoder maps images to latents and an entropy model controls bitrate through rate–distortion training [37, 1, 2, 29, 7]. Within this line, the variational autoencoder (VAE) provides a probabilistic view of latent bottlenecks through amortized inference and a KL-regularized evidence lower bound [22]. A range of VAE variants improve latent expressiveness or disentanglement, including β-VAE [18], hierarchical inference such as Ladder VAE [36], and flow-based posteriors that relax restrictive Gaussian assumptions [31]. Discrete-latent VAEs such as VQ-VAE connect generative modeling with quantized codes, and are often used when explicit codebooks are desired [38]. More recently, latent diffusion models have shown that strong generative priors can be trained in a compressed latent space defined by an autoencoder [32]. These latent generative models are

powerful, but they are not designed for scientific compression where the primary goal is faithful reconstruction of multi-field tensors under strict distortion control, and where errors concentrate around structured discontinuities rather than perceptual texture.

2.2 Compression for Earth system data

Earth system datasets pose a distinct compression regime because they are high-dimensional, multi-variable, and tightly coupled to scientific objectives and downstream modeling. In practice, the community has long relied on *error-bounded* scientific compressors to reduce storage and I/O overhead in HPC workflows, with ZFP and SZ as representative examples [24, 9]. Recent work has started to explore neural representations as a bridge between compression and learning [25, 17]. These efforts suggest that learned compression can move beyond storage reduction and begin to preserve task-relevant structure. While effective for file size reduction, such compressors are primarily value-oriented and do not produce a semantic representation that can be reused by learning systems. As a result, they rarely reduce the end-to-end cost of downstream tasks. In contrast, DC-Ocean targets compression as a reusable latent state for downstream modeling, enabling forecasting directly in compressed space rather than treating compression as a terminal storage operation.

2.3 Data-driven ocean forecasting and its horizon constraints

Data-driven ocean forecasting has progressed rapidly, yet most systems are evaluated in the short-range regime. WenHai provides an eddy-resolving global forecast system and reports skill up to a 10-day lead time for key variables [8]. FuXi-Ocean targets sub-daily global ocean forecasting and designs its architecture to handle complex multi-variable diurnal-cycle dynamics that emerge at intra-day resolution [19]. Benchmarks such as OceanBench further reflect this community focus by standardizing evaluation tracks for global short-range prediction, typically spanning 1–10 days [11]. A common design choice in these high-resolution systems is a ViT-style backbone, where the initial condition is partitioned into patches and mapped into tokens for global modeling [10]. This tokenization is often a practical response to extreme spatial resolution, but it behaves like a coarse and lossy downsampling at the very first layer. Even when short-range skill is strong, full-resolution global training remains dominated by data movement. I/O is slow, and both training and inference are expensive at $1/12^\circ$ scale. In our profiling, more than two-thirds of total training time is spent on reading and staging data rather than on GPU computation. These costs make long-horizon experiments difficult, and they limit the practical use of multi-step rollouts at high resolution.

3 Method

3.1 Problem Formulation.

We propose DC-Ocean, a deep latent compression framework for large-scale, high-resolution global ocean fields. Given the ocean state at the current time step, $\mathbf{X}_t \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of physical variables or input channels, and H and W denote the spatial grid dimensions along latitude and longitude, respectively, DC-Ocean learns a compact latent representation that enables accurate reconstruction of the input while remaining suitable for downstream modeling in latent space. An encoder $\mathcal{E}$ maps the physical field into a compressed representation $\mathbf{Z}_t = \mathcal{E}(\mathbf{X}_t)$, where $\mathbf{Z}_t \in \mathbb{R}^{(\alpha C)\times \frac{H}{r}\times \frac{W}{r}}$, corresponding to a spatial compression factor of r along both latitude and longitude dimensions and a channel expansion factor α, where α denotes the number of latent channels used to represent each input channel. A decoder $\mathcal{D}$ reconstructs the input state as $\hat{\mathbf{X}}_t = \mathcal{D}(\mathbf{Z}_t)$, recovering the original spatial resolution. Separately, downstream prediction models operate directly on the latent representation to forecast future latent states $\mathbf{Z}_{t+1}, \ldots, \mathbf{Z}_{t+n}$, which are decoded to the physical space using the same decoder $\mathcal{D}$, yielding future ocean states $\hat{\mathbf{X}}_{t+k} = \mathcal{D}(\mathbf{Z}_{t+k})$ for $k = 1, \ldots, n$.

We consider five fundamental ocean state variables that characterize the physical state of the global ocean, including temperature (T), salinity (S), zonal and meridional components of ocean currents (U and V), and sea surface height (SSH). While SSH is defined only at the surface, the remaining four variables exhibit substantial vertical variability and are represented across multiple depth levels. We discretize the water column into 26 depth layers spanning from the surface to 1500 m: 0, 2,

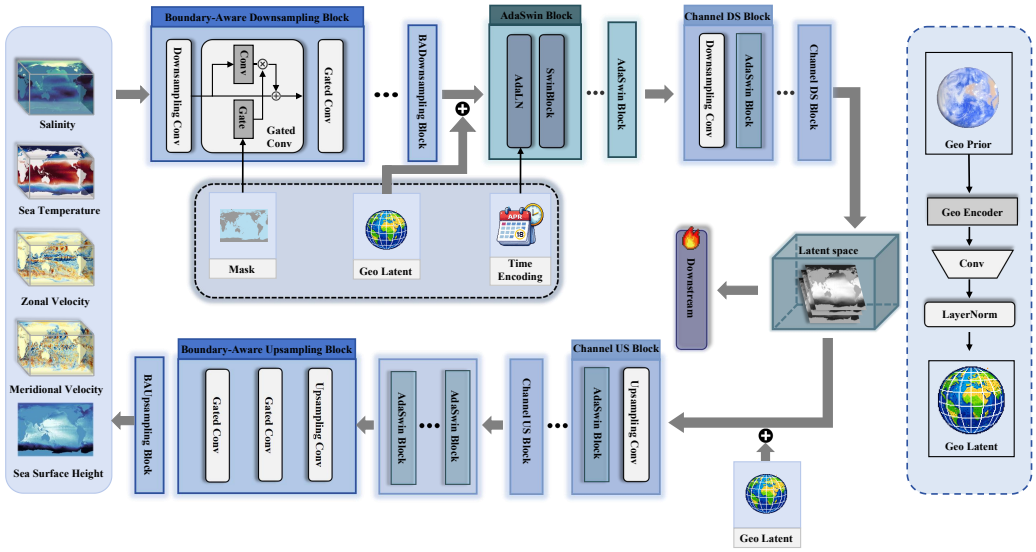

Figure 2: **Overall framework of DC-Ocean.** DC-Ocean consists of three core types of modules that operate along complementary dimensions. *Boundary-Aware Downsampling Blocks* and *Boundary-Aware Upsampling Blocks* are responsible for spatial resolution scaling. *Adaptive Swin Transformer Blocks (AdaSwin Blocks)* serve as the backbone of the network for extracting key spatiotemporal features. *Channel Downsampling Blocks (Channel DS Blocks)* adjust representation size along the channel dimension. Forecasting is performed directly in the latent space, and the predicted latent representations are decoded by a pretrained decoder.

4, 6, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 200, 300, 400, 500, 600, 800, 1000, 1250, and 1500 m. Consequently, the complete ocean state comprises a total of $C = 105$ channels, corresponding to 4×26 subsurface variable channels together with one surface SSH channel.

3.2 Model Architecture

To tackle the challenges of ocean data compression discussed above, DC-Ocean is designed to inject fixed geographic structure and temporal information into the latent representation, remove spatial redundancy through early downsampling while learning spatiotemporal features on a reduced grid, and explicitly model fixed land–sea discontinuities at both the value and feature levels to enable stable reconstruction near coastlines. We describe each part below.

3.2.1 Geographic Priors and Temporal Conditioning

Global ocean states are governed by fixed geographic structure and seasonal variability. We therefore incorporate lightweight geographic priors and temporal conditioning into DC-Ocean.

Geographic priors. We inject known, time-invariant geographic information, including spherical latitude–longitude encodings, coarse ocean basin indicators, and a fixed land–sea mask, to expose global position, regional context, and coastline structure.

Temporal conditioning. To account for temporal variability, we encode the observation date as a temporal embedding and apply it via Adaptive Layer Normalization (AdaLN) to condition the reconstruction process [15].

3.2.2 Overall Architecture

DC-Ocean adopts a streamlined latent compression pipeline guided by two principles: *removing redundancy* and *feature extraction*. Although global ocean fields are defined on extremely large grids,

their strong spatial continuity leads to substantial redundancy at full resolution. We therefore apply lightweight spatial downsampling early to reach a compact grid, and concentrate representational learning on this low-resolution space.

Boundary-aware spatial scaling. DC-Ocean adjusts spatial resolution in a staged manner using Boundary-Aware Downsampling and Upsampling Blocks. Each stage performs lightweight convolution for spatial scaling, followed by gated convolutions that incorporate the fixed land–sea mask. During downsampling, channel dimensionality is progressively increased to compensate for spatial reduction. During decoding, spatial resolution is restored while channel dimensionality is reduced to support accurate reconstruction.

Feature extraction with AdaSwin blocks. DC-Ocean employs AdaSwin Blocks as the primary modules for feature extraction. Each block is based on the Swin Transformer architecture [26], consisting of window-based self-attention and feed-forward networks. AdaLN is integrated into the blocks and conditioned on temporal embeddings, allowing the model to capture time-dependent characteristics in ocean states.

Channel-wise compression and expansion. DC-Ocean adjusts representation size by compressing and expanding the channel dimension to reach the desired latent dimensionality. Channel Downsampling and Upsampling Blocks are applied while keeping spatial resolution fixed, enabling precise control over feature dimensionality. Each channel scaling operation is followed by AdaSwin Blocks, which continue to extract and refine informative features at a constant spatial scale.

3.2.3 Boundary-aware Modeling and Interpolation

Unlike natural images, where discontinuities vary across samples, land–sea boundaries in ocean data are geographically determined and fixed over time. To address this mismatch, DC-Ocean adopts a boundary-aware strategy that operates at both the value level and the feature level, enabling stable reconstruction near coastlines.

Spherical Geodesic Front Propagation (SGFP) Interpolation. SGFP performs interpolation exclusively over land regions and does not participate in model training or introduce any additional supervision. Its role is to regularize values near fixed land–sea boundaries by providing smooth and stable transitions in coastal areas, thereby stabilizing subsequent convolutional processing. The progressive interpolation behavior near coastlines is illustrated in Figure 8, while the complete algorithmic details are provided in Appendix G.

Boundary-aware Gated Convolution. Unlike natural images, land regions in ocean data correspond to fixed geographic masks and are excluded from supervision, making naive convolution prone to feature mixing across coastlines. Inspired by gated convolution for image inpainting [39], we introduce a boundary-aware gated convolution that conditions feature aggregation on the land–sea mask.

Given an input feature map $\mathbf{x}$ and a binary land–sea mask $\mathbf{m}$, we compute a gated residual update:

$$\mathbf{y} = \mathbf{x} \; + \; \Big(\phi(\mathbf{W}_f * \mathbf{x})\Big) \odot \Big(\sigma(\mathbf{W}_g * \mathbf{m})\Big), \tag{1}$$

where $*$ denotes convolution, $\mathbf{W}_f$ and $\mathbf{W}_g$ are learnable kernels, $\phi(\cdot)$ is a nonlinear activation, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes element-wise multiplication.

3.3 Training Strategy

3.3.1 Variable-wise Training Strategy

Global ocean fields contain heterogeneous variables with distinct spatial characteristics, making joint compression across variables unstable: reconstruction errors evolve inconsistently, with some variables improving while others degrade (Figure 5). In contrast, jointly compressing multiple depth levels of the same variable is consistently beneficial, yielding lower reconstruction error than depth-wise independent models and indicating that cross-depth sharing better exploits redundancy.

Therefore, we adopt a variable-wise training strategy: T, S, and U are modeled separately across all depth levels, while SSH is jointly modeled with V, which yields the lowest reconstruction error. Additional details are provided in Appendix D.

3.3.2 Training Loss.

We adopt a latitude-weighted reconstruction loss and further aggregate channel-wise errors using a softmax weighting. For each channel c, we first compute a latitude-weighted RMSE over ocean grid points:

$$\mathrm{RMSE}_c = \sqrt{\frac{\sum_{i=1}^{H}\sum_{j=1}^{W} \alpha_i \, m_{c,i,j} \left(\hat{X}_{c,i,j} - X_{c,i,j}\right)^2}{\sum_{i=1}^{H}\sum_{j=1}^{W} m_{c,i,j}}} \tag{2}$$

where α_i denotes the latitude-dependent area weight and $m_{c,i,j}$ is the land–sea mask indicating ocean grid points.

We then obtain normalized channel weights via a softmax over channel-wise RMSE values and compute the final reconstruction loss as

$$w_c = \frac{\exp(\mathrm{RMSE}_c)}{\sum_{c'=1}^{C} \exp(\mathrm{RMSE}_{c'})}, \quad \mathcal{L}_{\mathrm{rec}} = \sum_{c=1}^{C} w_c \, \mathrm{RMSE}_c.$$

4 Experiment

4.1 Experimental Setup

4.1.1 Dataset Configuration

HYCOM-RD. We train and evaluate DC-Ocean using the HYCOM Reanalysis Dataset, a global ocean dataset with 6-hour temporal resolution. The dataset provides gridded ocean states at $1/12^\circ$ horizontal resolution and is widely used in data-driven ocean modeling.

Following standard practice, we use approximately 8.5 years of data (January 2006 to June 2014) for training, a six-month period (July to December 2014) for validation, and one year (January to December 2015) for testing. All experiments are conducted at a 6-hour sampling interval.

4.1.2 Implementation Details

DC-Ocean is implemented in PyTorch and trained using the AdamW optimizer [30, 21, 28]. The compression model is optimized using AdamW with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. We adopt a cosine annealing learning rate schedule with linear warmup: the learning rate is linearly increased from 1×10^{-5} to the base learning rate over the first 5,000 steps, and then decayed following a cosine schedule to 5×10^{-6} over a total of 5×10^4 training steps.

Training is conducted using mixed-precision with `bf16`. All experiments are performed on 8 NVIDIA H200 GPUs with a batch size of 1 per GPU. The total training time is approximately 142 hours. End-to-end inference and evaluation on the full 2015 dataset takes approximately 3 minutes.

The input ocean fields have a spatial resolution of H = 2160 and W = 4320. A downsampling factor of r = 24 is applied independently to each spatial dimension, resulting in a latent spatial resolution of 90×180, with a channel expansion factor of $\alpha = 18$.

4.1.3 Evaluation Metrics

RMSE. We evaluate reconstruction accuracy using the latitude-weighted root mean squared error (RMSE) defined in Eq. 2, following standard practice in global ocean forecast evaluation [8].

nRMSE. To highlight relative improvements, we report the normalized RMSE (nRMSE), defined as the RMSE difference normalized by the best-performing DC-Ocean configuration reported in Table 1.

$$\mathrm{nRMSE} = \frac{\mathrm{RMSE} - \mathrm{RMSE}_{\text{DC-Ocean}}}{\mathrm{RMSE}_{\text{DC-Ocean}}}.$$

SEDI. To evaluate performance on extreme events, we report the Symmetric Extremal Dependence Index (SEDI) [12], a standard skill score designed for rare-event forecasting. SEDI is computed from the hit rate H and false alarm rate F derived from thresholded extreme-event indicators over ocean grid points, with latitude-based area weighting. The score is defined as:

$$\mathrm{SEDI} = \frac{\log F - \log H - \log(1-F) + \log(1-H)}{\log F + \log H + \log(1-F) + \log(1-H)}.$$

Depth (m)	S	T	U	V	SSH
0	0.0411	0.1486	0.00911	0.0088	0.0084
10	0.0383	0.1423	0.0084	0.0082	–
50	0.0417	0.1979	0.0129	0.0124	–
100	0.0412	0.2051	0.0147	0.0144	–
150	0.0493	0.2418	0.0183	0.0183	–
200	0.0455	0.2130	0.0181	0.0183	–
300	0.0403	0.1797	0.0170	0.0171	–
400	0.0366	0.1629	0.0161	0.0162	–
500	0.0341	0.1521	0.0154	0.0155	–
600	0.0310	0.1463	0.0156	0.0158	–
800	0.0260	0.1317	0.0164	0.0165	–
1000	0.0220	0.1085	0.0144	0.0143	–
1250	0.0187	0.08912	0.0127	0.0125	–
1500	0.0193	0.08724	0.0129	0.0127	–
Average	**0.0371**	**0.1645**	**0.0128**	**0.0128**	**0.0084**

Table 1: Reconstruction RMSE at different depth levels.

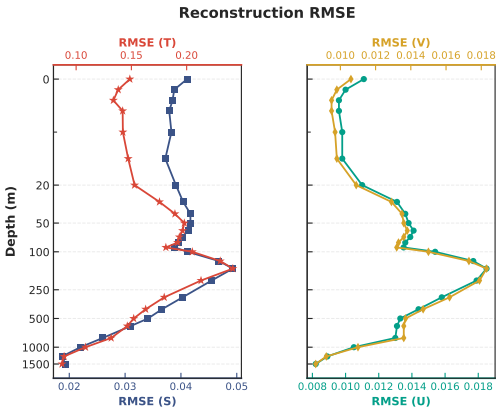

Figure 3: Vertical profiles of reconstruction RMSE across depth levels for different ocean variables.

4.2 Reconstruction Performance

To evaluate reconstruction performance, Table 1 reports the latitude-weighted RMSE of DC-Ocean at representative depths and averaged over all 26 depth levels. Across all ocean variables, DC-Ocean achieves consistently low reconstruction errors, indicating that the learned latent representations effectively remove redundant information while preserving the essential structures required for faithful recovery. Such low reconstruction errors provide a reliable latent state for downstream forecasting.

Figure 3 shows the vertical distribution of reconstruction errors for representative variables. Across variables, the error exhibits a consistent depth-dependent pattern: it increases from the surface into the upper ocean, reaches a maximum around 150–200m, and gradually decreases in deeper layers. This depth range corresponds to the thermocline, a well-known natural oceanographic feature. The errors in this region remain controlled and vary smoothly with depth, indicating systematic rather than abrupt behavior.

4.3 Comparison Results

We compare DC-Ocean with Deep Compression Autoencoder (DCAE) [6], a compression framework originally developed for high-resolution image diffusion models. To ensure a fair comparison, all baselines including DCAE are adapted to our ocean data setting; these modifications are described in detail in Appendix C.

As shown in Table 2, both DCAE variants exhibit substantially higher reconstruction errors across all evaluated variables than DC-Ocean, despite using significantly fewer parameters (Table 4). This gap is further confirmed by SEDI analysis (Figure 4), where DCAE consistently underperforms at both low and high quantiles. Notably, DC-Ocean operates with single-variable, single-depth inputs, indicating that its gains arise from more effective latent compression rather than increased input complexity.

We additionally evaluated generic AE, VAE, and VQ-VAE baselines, which achieve reasonable compression but show reconstruction errors higher than DC-Ocean and comparable to DCAE. Overall, these results highlight the challenge of directly transferring generic image compression models to large-scale ocean fields under high downsampling factors, and the advantage of our domain-adapted latent compression approach.

Method	Reconstruction Variables				
	S0	T0	U0	V0	SSH
AE	0.66	0.422	2.83	2.89	1.55
VAE	7.73	5.57	6.54	5.79	8.13
VQ-VAE	1.62	1.085	4.503	4.33	3.60
DCAE–sample	6.558	5.538	4.700	4.333	3.789
DCAE	1.415	0.4798	3.300	3.066	2.000
SV–SD	0.130	0.150	1.867	1.967	0.947
SV–RD	0.442	0.348	2.567	2.567	0.947
RV–RD	0.675	0.253	2.933	2.933	1.526
MV–MD	0.468	0.125	1.600	1.533	0.947
Spatial-Heavy	0.844	0.377	3.500	3.433	1.789
w/o Gated+SGFP	0.208	0.143	2.033	2.067	1.000
w/o weighted loss	0.391	0.185	2.130	2.100	1.095
w/o geo-temporal embeddings	0.369	0.177	2.300	2.330	1.022

Table 2: Normalized reconstruction error (nRMSE) relative to DC-Ocean. Lower is better.

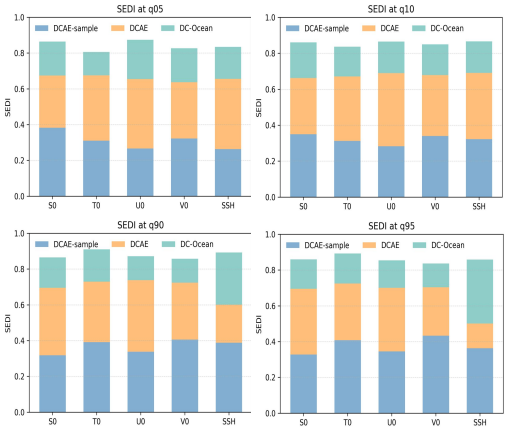

Figure 4: SEDI comparison at different quantiles (5%, 10%, 90%, and 95%), where higher values indicate better performance.

4.4 Ablation Studies

4.4.1 Training Strategy

We next examine the impact of different input organization strategies during training. We use **S/M** to denote single/multiple inputs, **V/D** for variable/depth, and **R** for random selection. We consider multiple input organization strategies. The exact input formats associated with each strategy are detailed in Table 4.

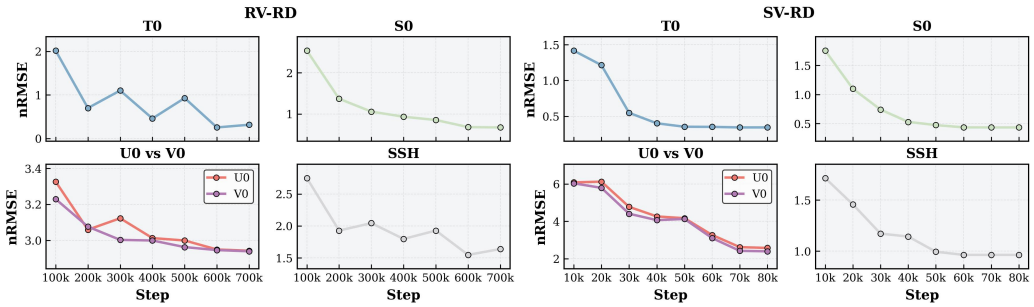

Figure 5: Training dynamics under different training strategies. Top: nRMSE evolution for temperature (T0) and salinity (S0). Bottom: velocity components (U0/V0) and SSH.

Figure 5 shows the training dynamics. Sharing parameters across heterogeneous variables destabilizes optimization and slows convergence, likely due to interference from variable-specific spatial patterns. In contrast, training multiple depth levels of the same variable is stable, as redundancy across depth layers can be exploited without conflicting patterns. These trends are consistent with Table 2, where single-variable, multi-depth training achieves the lowest reconstruction error, highlighting that leveraging cross-depth redundancy is crucial for effective compression.

4.4.2 Model Architecture

Spatial-Heavy. The *Spatial-Heavy* variant shifts capacity from latent feature extraction to spatial scaling by reducing AdaSwin blocks and increasing convolutions, keeping a similar parameter count (Table 4). As shown in Table 2, this reduces reconstruction accuracy, indicating that for large-scale ocean fields, preserving capacity at the compressed resolution is more important than emphasizing spatial scaling. The comparable parameter budget further suggests that this performance gap mainly arises from capacity allocation rather than model size.

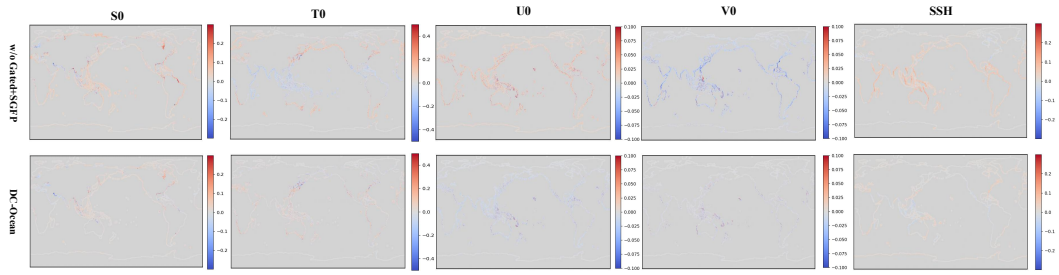

Figure 6: Visualization of reconstruction errors near land–sea boundaries under architectural ablations. The colorbar represents RMSE values.

Without GatedConv + SGFP. Removing Gated Convolution and SGFP moderately increases RMSE (Table 2) but causes pronounced errors and artifacts along land–sea boundaries (Figure 6), highlighting the importance of explicit boundary modeling for stable reconstruction near geographic discontinuities.

4.5 Latent-Space Forecasting

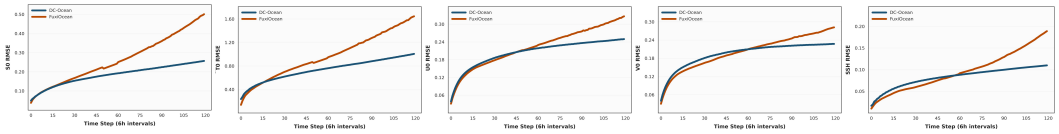

Figure 7: RMSE over forecast lead time (6-hour intervals, up to 120 steps) for DC-Ocean and FuXi-Ocean.

We evaluate DC-Ocean for latent-space forecasting by predicting latent states at 6-hour intervals over a 30-day horizon, corresponding to lead times up to t+120. Detailed experimental and implementation settings are provided in Appendix F.

Beyond forecast accuracy, we assess the practical efficiency of the resulting forecasting system. Under a fair comparison, DC-Ocean and FuXi-Ocean are evaluated on the same hardware configuration (4 $\times$ H100) with comparable model capacities: 1520.142M parameters for DC-Ocean and 1557.722M for FuXi-Ocean. DC-Ocean reduces the end-to-end data I/O time per temporal snapshot from 8.0 s to 0.42 s and shortens the overall training time from 81 hours to 36 hours. Although storage reduction is not the primary objective of DC-Ocean, the latent representation decreases the dataset footprint from 28 TB to 850 GB.

The latent representation is stored as fixed-shape dense tensors that can be consumed directly by downstream forecasting models. In the default configuration, the four variable-group latents are concatenated into a full-state tensor of shape (1872,90,180) at each time step. The latent dataset is stored in FP16 using Zarr with a chunk shape of (1,468,90,180). Blosc with LZ4 is used as the lossless storage backend, with compression level 5 and byte shuffling. This compression is applied only at the container level: DC-Ocean does not use scalar or vector quantization, a learned entropy model, or arithmetic coding. Consequently, both the original physical fields and the latent states remain chunked multidimensional arrays that are directly compatible with standard data-loading pipelines. The reported storage figures therefore correspond to dense latent arrays rather than entropy-coded archival bitstreams.

We finally examine the forecasting accuracy in latent space. Figure 7 presents the RMSE evolution over lead time for five representative variables. Compared with direct forecasting on full-resolution fields, for which autoregressive errors accumulate rapidly, latent-space forecasting with DC-Ocean exhibits substantially slower error growth. It achieves accuracy comparable to FuXi-Ocean at short lead times and consistently lower RMSE in the medium- and long-range regimes.

5 Conclusion

We present DC-Ocean, the first deep latent compression framework for multi-variable, multi-depth global $1/12^\circ$ ocean states. DC-Ocean is tailored to the unique challenges of ocean data, including extreme spatial resolution, heterogeneous physical variables across depth, and sharp land–sea discontinuities. By learning compact latent representations that preserve physically meaningful structures under aggressive downsampling, DC-Ocean achieves high-fidelity reconstruction while remaining suitable for downstream tasks.

Beyond compression, DC-Ocean enables an effective pathway toward long-horizon forecasting on global high-resolution grids. Forecasting directly in the learned latent space substantially reduces I/O and computational overhead and, more importantly, slows error accumulation during autoregressive rollouts. As a result, latent-space forecasting achieves competitive accuracy at short lead times and increasingly favorable performance at medium and long horizons. Together, these results demonstrate that latent compression can serve as a reusable state representation, making long-horizon autoregressive prediction on massive global ocean fields both feasible and reliable.

Acknowledgments and Disclosure of Funding

This work was supported by AI for Science Program (ID: 2025-GZL-RGZN-BTBX-02017), Shanghai Municipal Commission of Economy and Information.

References

[1] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR)*, 2017.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *International Conference on Learning Representations (ICLR)*, 2018.

[3] Neil Barton, E Joseph Metzger, Carolyn A Reynolds, Benjamin Ruston, Clark Rowley, Ole Martin Smedstad, James A Ridout, Alan Wallcraft, Sergey Frolov, Patrick Hogan, et al. The navy's earth system prediction capability: A new global coupled atmosphere-ocean-sea ice prediction system designed for daily to subseasonal forecasting. *Earth and Space science*, 8(4):e2020EA001199, 2021.

[4] EW Blockley, MJ Martin, AJ McLaren, AG Ryan, J Waters, DJ Lea, I Mirouze, KA Peterson, A Sellar, and D Storkey. Recent development of the met office operational ocean forecasting system: an overview and assessment of the new global foam forecasts. *Geoscientific Model Development*, 7(6):2613–2638, 2014.

[5] Eric P. Chassignet, Harley E. Hurlburt, Ole Martin Smedstad, George R. Halliwell, Patrick J. Hogan, Alan J. Wallcraft, Remy Baraille, and Rainer Bleck. The hycom (hybrid coordinate ocean model) data assimilative system. *Journal of Marine Systems*, 65(1):60–83, 2007. Marine Environmental Monitoring and Prediction.

[6] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models, 2025.

[7] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] Yingzhe Cui, Ruohan Wu, Xiang Zhang, Ziqi Zhu, Bo Liu, Jun Shi, Junshi Chen, Hailong Liu, Shenghui Zhou, Liang Su, Zhao Jing, Hong An, and Lixin Wu. Forecasting the eddying ocean with a deep neural network. *Nature Communications*, 16:2268, 2025.

[9] Sheng Di and Franck Cappello. Fast error-bounded lossy hpc data compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2016.

[10] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[11] Anass El Aouni, Quentin Gaudel, J. Emmanuel Johnson, Charly Regnier, Julien Le Sommer, Simon van Gennip, Ronan Fablet, Marie Drevillon, Yann Drillet, and Pierre-Yves Le Traon. Oceanbench: A benchmark for data-driven global ocean forecasting systems, 2025. NeurIPS 2025 poster (Datasets & Benchmarks track).

[12] Christopher AT Ferro and David B Stephenson. Extremal dependence indices: Improved verification measures for deterministic forecasts of rare binary events. *Weather and Forecasting*, 26(5):699–713, 2011.

[13] Zulema D Garraffo, James A Cummings, S Paturi, Y Hao, D Iredell, T Spindler, et al. Rtofs-da: real time ocean-sea ice coupled three dimensional variational global data assimilative ocean forecast system. *Research activities in Earth system modelling*, 2020.

[14] Stephen M Griffies, Matthew J Harrison, Ronald C Pacanowski, Anthony Rosati, et al. A technical guide to mom4. *GFDL Ocean Group Tech. Rep*, 5(5):371, 2004.

[15] Yunhui Guo, Chaofeng Wang, Stella X. Yu, Frank McKenna, and Kincho H. Law. Adaln: A vision transformer for multidomain learning and predisaster building information extraction from images. *Journal of Computing in Civil Engineering*, 36(5):04022024, 2022.

[16] Madec Gurvan, Romain Bourdallé-Badie, Clément Bricaud, Diego Bruciaferri, Daley Calvert, Jérôme Chanut, Emanuela Clementi, Andrew Coward, Damiano Delrosso, Christian Ethé, et al. Nemo ocean engine. 2017.

[17] Tao Han, Zhenghao Chen, Song Guo, Wanghan Xu, Wanli Ouyang, Lei Bai, et al. Climate science data can be compressed efficiently by dual-stage extreme compression with a variational auto-encoder transformer. *Communications Earth & Environment*, 6:955, 2025.

[18] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

[19] Qiusheng Huang, Yuan Niu, Xiaohui Zhong, Anboyu Guo, Lei Chen, Dianjun Zhang, Xuefeng Zhang, and Hao Li. Fuxi-ocean: A global ocean forecasting system with sub-daily resolution, 2025. NeurIPS 2025 oral presentation.

[20] Lellouche Jean-Michel, Greiner Eric, Bourdallé-Badie Romain, Garric Gilles, Melet Angélique, Drévillon Marie, Bricaud Clément, Hamon Mathieu, Le Galloudec Olivier, Regnier Charly, et al. The copernicus global 1/12 oceanic and sea ice glorys12 reanalysis. *Frontiers in Earth Science*, 9:698876, 2021.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[22] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[23] Jean-Michel Lellouche, Eric Greiner, Olivier Le Galloudec, Charly Regnier, Mounir Benkiran, Charles-Emmanuel Testut, Romain Bourdalle-Badie, Marie Drevillon, Gilles Garric, and Yann Drillet. The mercator ocean global high-resolution monitoring and forecasting system. *New Frontiers in Operational Oceanography*, 1:563–592, 2018.

[24] Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014.

[25] Jinyang Liu, Sheng Di, Kai Zhao, Sian Jin, Dingwen Tao, Xin Liang, Zizhong Chen, and Franck Cappello. Exploring autoencoder-based error-bounded compression for scientific data. In *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, 2021.

[26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019, June 2022.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[29] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[31] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, 2015.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] Andreas Schiller, Gary B Brassington, Peter Oke, Madeleine Cahill, Prasanth Divakaran, Mikhail Entel, Justin Freeman, David Griffin, Mike Herzfeld, Ron Hoeke, et al. Bluelink ocean forecasting australia: 15 years of operational ocean service delivery with societal, economic and environmental benefits. *Journal of Operational Oceanography*, 13(1):1–18, 2020.

[34] James A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[35] Gregory C Smith, François Roy, Mateusz Reszka, Dorina Surcel Colan, Zhongjie He, Daniel Deacu, Jean-Marc Belanger, Sergey Skachko, Yimin Liu, Frédéric Dupont, et al. Sea ice forecast verification in the canadian global ice ocean prediction system. *Quarterly Journal of the Royal Meteorological Society*, 142(695):659–671, 2016.

[36] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[37] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017.

[38] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[39] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019.

A Limitations

DC-Ocean is currently evaluated only on autoregressive ocean forecasting. While the learned latent representations prove effective for this task, their applicability to other ocean-related downstream tasks, such as anomaly detection, feature extraction, or coupled modeling, remains unexplored. Investigating these applications could reveal additional strengths and limitations of the compressed latent space and represents an important direction for future work.

Our evaluation is also limited to two high-resolution reanalysis datasets, HYCOM and GLORYS, five ocean variables, and 26 depth levels down to 1500 m. Generalization to other operational data products, resolutions, deeper layers, and additional ocean variables remains unverified.

Furthermore, the proposed latent compression framework could potentially be extended to other geoscientific domains, including atmospheric or climate modeling. Evaluating its performance in these settings would help assess the generality and versatility of the approach beyond oceanographic data.

B Broader Impact

This work aims to improve the efficiency and scalability of data-driven ocean modeling by enabling compact and predictive latent representations of high-resolution ocean states. By reducing data storage, I/O overhead, and computational cost, DC-Ocean has the potential to facilitate longer-horizon and higher-resolution forecasting in scientific and operational settings, including climate research and ocean monitoring. More broadly, representing large-scale ocean states in a compressed latent space while retaining predictive capability may provide a general and efficient paradigm for applying downstream learning tasks to high-resolution ocean data, and could become an important approach for handling large-scale ocean imagery in future data-driven applications.

The proposed method is purely data-driven and does not introduce new sources of environmental risk or societal harm. However, as with all data-driven forecasting approaches, its outputs should be interpreted as complementary tools rather than replacements for physics-based models or expert judgment. Responsible use requires careful validation and domain-aware interpretation, particularly when applied to decision-making contexts.

C Baseline Adaptation Details

All baseline models are adapted to the ocean compression setting with minimal modifications, ensuring fair comparison with DC-Ocean. Each baseline uses the same land-masked reconstruction loss as DC-Ocean, excluding land regions from the loss computation, and is adjusted to handle the ocean-specific input and output formats.

We evaluate the following baselines:

Deep Compression Autoencoder (DCAE) [6] is adapted under two configurations. **DCAE-sample** follows the sampling-based setting with adversarial training enabled and a KL divergence weight of 10^{-6}. **DCAE** adopts a deterministic setting where latent sampling and adversarial loss are disabled. To keep the KL regularization scale consistent with DC-Ocean (10^{-12}), the KL weight of the deterministic DCAE variant is set to 10^{-12}. The core architecture and training scheme of DCAE are retained, with only input dimensionality and land–sea masking adapted for ocean data.

Autoencoder (AE), **Variational Autoencoder (VAE)**, and **Vector-Quantized VAE (VQ-VAE)** are also adapted to our input/output formats. Their reconstruction loss is replaced with the same land-masked loss used for DC-Ocean. Apart from these adjustments, their architectures and training schemes follow standard implementations.

These adaptations ensure that all baselines operate under comparable conditions, allowing a fair assessment of compression performance across different latent representation methods.

D Additional Details on Joint Compression with SSH

Variable	SSH RMSE
S	0.02235
T	0.04871
U	0.00913
V	**0.00842**

Table 3: Reconstruction RMSE of SSH under different joint-compression settings, where SSH is jointly trained with a single variable.

Table 3 reports the reconstruction RMSE of SSH under different joint-compression settings. Among all configurations, SSH achieves its lowest reconstruction error when jointly compressed with V, suggesting that V provides the most compatible representation for joint compression with SSH.

We further observe that the reconstruction error of SSH exhibits a positive correlation with the overall RMSE of the jointly trained variable. This behavior is a direct consequence of our loss formulation, which balances reconstruction errors across channels and prevents any single variable from dominating the optimization. As a result, the reconstruction quality of SSH tends to track the aggregate error level of the corresponding joint-compression setting.

When SSH is jointly trained with V, the input consists of 27 channels. However, the dimensionality of the intermediate latent representation is kept identical to that of the four-variable setting. Specifically, the number of latent channels is fixed to $26 \times \alpha$, corresponding to 26 depth levels multiplied by the expansion factor α, yielding a latent tensor of shape $(26\alpha, H/r, W/r)$. That is, the latent representation does not scale with the input channel count, and the same intermediate feature format is used across all variable-wise training configurations.

E Experimental Parameter Count

Method	Input Setting	Input Size	Models Required	Params / Model	Total Params
DC-Ocean	SV–MD	(D, H, W)	4	601,525,447	2,406,101,788
DCAE	SV–SD	$(1, H, W)$	105	200,527,459	21,055,383,195
DCAE-sample	SV–SD	$(1, H, W)$	105	200,527,459	21,055,383,195
SV–SD	SV–SD	$(1, H, W)$	105	120,124,123	12,613,032,915
SV–RD	SV–RD	$(1, H, W)$	4	601,525,447	2,406,101,788
RV–RD	RV–RD	$(1, H, W)$	1	3,059,160,289	3,059,160,289
MV–MD	MV–MD	$(V \times D, H, W)$	1	3,059,160,289	3,059,160,289
Spatial-Heavy	SV–SD	$(1, H, W)$	105	136,977,379	14,382,624,795
w/o Gated+SGFP	SV–SD	$(1, H, W)$	105	119,824,353	12,581,556,065

Table 4: Parameter count comparison under full-variable, full-depth coverage. All models take three-dimensional inputs of shape (C, H, W). For random-input strategies, SV–RD randomly selects a single depth slice from the D channels of a given variable, whereas RV–RD randomly selects a single slice from the full $V \times D$ channel set. "Models Required" indicates the number of independently trained models needed to cover all variables and depth levels.

Table 4 reports the parameter count of the models used in our experiments to cover all variables and depth levels. Different methods adopt fundamentally different modeling strategies. The MV–MD baseline employs a single monolithic model that jointly processes all variables and depths. DC-Ocean adopts a grouped modeling strategy and requires four models to cover the full variable set. In contrast, single-variable and traditional baselines must train independent models for each variable–depth combination, resulting in a total of 105 models in our setting. Here, R-based variants apply random sampling along the variable or depth dimension during training, while sharing the same model weights across all variables or depths. Consequently, RV-based configurations are equivalent to the MV setting in terms of variable coverage, and RD-based configurations are equivalent to

the MD setting in terms of depth coverage; in both cases, the only difference lies in the channel dimensionality of the intermediate latent representations.

For fair system-level comparison, we report both the parameters of an individual model and the total number of parameters required to represent the complete ocean state. Overall, DC-Ocean achieves the lowest total parameter count among all methods, while also delivering the best performance in our experiments.

F Latent-Space Forecasting Setup

F.1 Forecasting Protocol

The original FuXi-Ocean forecaster consists of three components: (1) a large-kernel convolutional encoder that maps the full-resolution physical field to a latent tensor, (2) a Swin Transformer-based predictor that performs autoregressive forecasting in latent space, and (3) a large-kernel deconvolutional decoder that maps the predicted latent tensor back to physical space.

In our latent-space forecasting experiment, the Swin Transformer predictor is kept unchanged. Since the input is now a compressed latent representation produced by DC-Ocean rather than a full-resolution physical field, the original FuXi-Ocean input encoder is replaced by a lightweight two-layer MLP adaptor. All variables are concatenated along the channel dimension to form a tensor of size $1872 \times 90 \times 180$, and the adaptor projects this tensor to $1536 \times 90 \times 180$, matching the input format expected by the FuXi-Ocean predictor. The predicted DC-Ocean latents are subsequently decoded using the frozen DC-Ocean decoder for evaluation in physical space.

Training is conducted in two fully separated stages. First, the DC-Ocean autoencoder is trained and then completely frozen. The compressed latent states are generated offline and stored. Second, the forecasting model is trained on these fixed latents, with the MLP adaptor optimized jointly with the unchanged Swin predictor. The DC-Ocean encoder and decoder are not updated during forecasting.

Starting from time t, both models predict states at 6-hour intervals up to $t + 120$, corresponding to a 30-day forecast horizon. Both methods use the same training and validation periods, 120-step autoregressive rollout schedule, 2015 evaluation initial conditions, and four NVIDIA H100 GPUs. DC-Ocean latent forecasting uses a Charbonnier loss, whereas the full-resolution FuXi-Ocean baseline uses its latitude-weighted Charbonnier loss.

F.2 Fairness of the Comparison with FuXi-Ocean

The comparison is designed to isolate the effect of the input representation while keeping the forecasting backbone and evaluation protocol fixed. Specifically, the Swin Transformer predictor, optimization schedule, rollout length, data splits, evaluation initial conditions, and hardware configuration are shared by both methods. The only architectural change is the input/output mapping required to accommodate either the DC-Ocean latent representation or the original full-resolution physical field.

Table 5 summarizes the matched experimental settings. The reported parameter counts refer to trainable forecasting parameters. The frozen DC-Ocean encoder–decoder, which contains 2406.102M parameters, is excluded because it is pretrained once and reused across downstream forecasting experiments. Similarly, the reported 36-hour and 81-hour runtimes refer only to downstream forecast training; the one-time 142-hour DC-Ocean pretraining cost is reported separately.

This protocol keeps the forecasting backbone, optimization procedure, rollout schedule, evaluation conditions, and hardware fixed while changing only the representation and its associated input/output mapping. The resulting efficiency comparison therefore measures the benefit of reusable latent-space forecasting rather than a single end-to-end run that includes DC-Ocean autoencoder pretraining.

Setting	DC-Ocean latent forecasting	FuXi-Ocean forecasting
Forecast state	$1872 \times 90 \times 180$ latent tensor	$105 \times 2160 \times 4320$ physical tensor
Forecasting backbone	FuXi-Ocean Swin predictor	Same
Input/output mapping	Two-layer MLP adaptor; frozen DC-Ocean decoder	Original convolutional encoder and deconvolutional decoder
Training period	Jan. 2006–Jun. 2014	Same
Validation period	Jul.–Dec. 2014	Same
Loss	Charbonnier loss	Latitude-weighted Charbonnier loss
Rollout	6-hour autoregressive rollout, 120 steps	Same
Evaluation	2015 initial conditions and RMSE	Same
Hardware	4 NVIDIA H100 GPUs	Same
Trainable parameters	1520.142M	1557.722M
Forecast training time	36 h	81 h

Table 5: Comparison of the experimental settings for DC-Ocean latent forecasting and full-resolution FuXi-Ocean forecasting.

G Spherical Geodesic Front Propagation (SGFP)



Figure 8: Visualization of Spherical Geodesic Front Propagation (SGFP), showing the progressive inward propagation from coastlines following geodesic proximity.

Algorithm 1: Spherical Geodesic Front Propagation (SGFP)

Input: Ocean values v on grid; land–sea mask
Output: Interpolated values on land regions

Initialization:
Initialize ocean region Ω_o and land region Ω_l;

Preprocessing:
Compute distance-to-ocean $d_g(x)$ for all $x \in \Omega_l$;
Sort land points in ascending order of $d_g(x)$;

Front propagation:
for *each land point* $x \in \Omega_l$ *in sorted order* **do**
    Identify neighboring points $\mathcal{N}(x)$ that are already filled;
    Compute confidence weights $\gamma(y) = \exp(-d_g(y)/\tau), \quad y \in \mathcal{N}(x)$;
    Assign value $v(x) \leftarrow \frac{\sum_{y \in \mathcal{N}(x)} \gamma(y)\, v(y)}{\sum_{y \in \mathcal{N}(x)} \gamma(y)}$;

return *Interpolated field*

Algorithm 1 summarizes the complete Spherical Geodesic Front Propagation (SGFP) procedure. SGFP is a deterministic preprocessing algorithm applied once on a global latitude–longitude grid with a binary land–sea mask, where the ocean domain Ω_o contains valid physical values and the land domain Ω_l is undefined. The algorithm interpolates values only within Ω_l and does not participate

in model training. For each land point $x \in \Omega_l$, SGFP computes its spherical geodesic distance to the nearest ocean point and processes land points in ascending order of this distance. Specifically, the geodesic distance between two grid points with latitude–longitude coordinates (ϕ_1, λ_1) and (ϕ_2, λ_2) is computed using the Haversine formula:

$$d_g = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right),$$

where R denotes the Earth radius. At each propagation step, values are estimated from a local neighborhood of already-filled points. Both the neighborhood size and the distance-decay factor are configurable parameters; in our implementation, we use an 8-connected neighborhood and set $\tau = 4$. Neighboring points are aggregated using distance-based confidence weights $\gamma(y) = \exp(-d_g(y)/\tau)$.

G.1 Experimental Comparison

We compare SGFP with a standard padding strategy using boundary-region RMSE:

Method	**S0**	**T0**	**U0**	**V0**	**SSH**
SGFP	0.0713	0.1789	0.0992	0.0911	0.00922
Standard padding	0.0908	0.2142	0.1096	0.1106	0.0102
Improvement (%)	**21.5%**	**16.5%**	**9.5%**	**17.6%**	**9.6%**

Table 6: Comparison of boundary-region RMSE between SGFP and standard padding strategy. Lower values are better. Improvement percentage shows relative reduction achieved by SGFP.

SGFP consistently achieves lower boundary RMSE across all variables, demonstrating more effective boundary-aware modeling than padding-based alternatives. The improvement ranges from 9.5% to 21.5%, with particularly strong gains for S0 (21.5%) and V0 (17.6%).

H Region-Wise Reconstruction Evaluation

We partition the global ocean into four major basins: Pacific, Indian, Atlantic, and Southern Oceans. Table 7 shows that our method achieves low reconstruction RMSE across all basins.

Ocean Basin	**S0**	**T0**	**U0**	**V0**	**SSH**
Pacific	0.03493	0.1472	0.00911	0.00880	0.00741
Indian	0.04957	0.1606	0.00999	0.00910	0.00910
Atlantic	0.05013	0.1737	0.00101	0.00102	0.00920
Southern	0.04601	0.1271	0.00824	0.00791	0.01050

Table 7: Region-wise reconstruction RMSE across four major ocean basins. Lower values indicate better reconstruction performance.

I Additional Experiments on GLORYS Dataset

To further evaluate the generalization ability of our method, we conducted experiments on the GLORYS dataset [20], which provides global high-resolution ocean surface variables including sea surface temperature, salinity, and velocity fields, derived from satellite observations and reanalysis products.

To isolate the effect of spatial resolution from learned representations, we also construct a bicubic baseline.

Variable	RMSE
S0	0.0300
T0	0.0530
U0	0.0076
V0	0.0070
SSH	0.0030

Table 8: Reconstruction RMSE on GLORYS Dataset.

Day	S0	T0	U0	V0	SSH
1	0.059 / 0.082	0.106 / 0.129	0.029 / 0.024	0.028 / 0.024	0.009 / 0.007
3	0.117 / 0.160	0.235 / 0.267	0.059 / 0.053	0.056 / 0.050	0.022 / 0.018
10	0.226 / 0.274	0.516 / 0.543	0.108 / 0.108	0.105 / 0.104	0.047 / 0.045
15	0.260 / 0.328	0.614 / 0.677	0.128 / 0.137	0.123 / 0.132	0.061 / 0.061
30	0.333 / 0.430	0.869 / 0.980	0.175 / 0.183	0.170 / 0.172	0.080 / 0.087

Table 9: Forecasting RMSE on GLORYS Dataset (DC-Ocean / FuXi-Ocean).

Day	S0	T0	U0	V0	SSH
1	0.391	0.734	0.0336	0.0305	0.0364
3	0.414	0.771	0.0591	0.0545	0.0405
10	0.470	0.905	0.1128	0.1069	0.0594
15	0.507	1.012	0.1399	0.1334	0.0718
30	0.568	1.209	0.1897	0.1759	0.0969

Table 10: Forecasting RMSE using bicubic interpolation as a baseline.

As shown in Tables 8–10, our method achieves low reconstruction and forecasting RMSE on the GLORYS dataset, substantially outperforming the bicubic baseline. These results indicate that the gains are not solely due to reduced spatial resolution, but arise from the learned latent representation. Combined with previous results on the HYCOM dataset, this demonstrates that our method not only performs well on HYCOM, but also generalizes effectively to an independent dataset, highlighting robust predictive capability across different oceanographic data sources.

J Hyperparameter Sensitivity Analysis

We conduct a sensitivity analysis of key hyperparameters under a single-variable, single-depth setting. Here, c denotes the latent channel expansion factor, which controls the capacity of the latent representation, and d denotes the spatial downsampling factor, which affects the model's spatial resolution. Results are reported in terms of nRMSE.

Setting	c	d	S0	T0	U0	V0	SSH
Default	18	24	0.130	0.150	1.867	1.967	0.947
C_36	36	24	0.083	-0.089	1.684	1.670	0.143
C_9	9	24	0.575	0.351	3.260	3.340	1.855
D_6	18	6	-0.098	-0.129	0.980	1.050	-0.194
D_48	18	48	1.135	0.667	4.123	4.050	2.889

Table 11: Sensitivity analysis of latent channel expansion factor c and spatial downsampling factor d on model performance (nRMSE).

From Table 11, we select the latent channel expansion factor $c = 18$ and spatial downsampling factor $d = 24$, considering both model complexity and acceleration from downsampling while maintaining satisfactory reconstruction accuracy.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our main claims in the abstract , elaborate on the research scope and contributions in the introduction 1, summarize the paper's contributions at the end of the introduction 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated discussion of the papers limitations in the Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: this paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: as shown in Section 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We plan to release the code, data, and pretrained model checkpoints publicly upon acceptance of the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: as shown in Section 4

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: No, reporting error bars is too computationally expensive and is not standard in data-driven ocean forecasting.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We specify this part in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of the paper in Section B

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve any models or data with a high risk of misuse or safety concerns.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and pretrained models used in this work are publicly available and properly cited in the manuscript. License terms of the original assets are respected, and no proprietary or restricted-use materials are included.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: NA

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: NA

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.