We very much appreciate the reviewer's effort in the careful review and their time. THe following is our point-by-point response marked by blue. For better tracking, we also included screenshots of the relevant section in the revised text.

—---

Review #2
The authors have improved the paper, but the following point should still be clarified:

1. Authors rebuttal: “To provide further clarification, our assertion is that SST in simulations is unaffected by random noise and measurement errors commonly associated with data collection platforms and corruption.”

The model SST is affected by modeling error. You are just sidestepping this by declaring the model results as ground truth in your validation (which is fine for a first test). This should be clarified in the manuscript.

We agree with the reviewer that as a first test we ignored the influence of measurement errors and model errors. This was our intention as a demonstration of applying MAE on geophysical signals. We made it clear now in the text. The following sentence was added to line 93:

95 generalize to unseen data of a different spatial resolution. These fields do not have additional added noise and errors but can contain deviations from reality due to model imperfection.

2. Thank you for clarifying the origin of the MAE implementation. Please include the https link to the Github repository that you used.

We added the reference to the original MAE code explicitly in line 74.

75 Figure 1 shows an example of the MAESSTRO architecture derived from the original MAE open-source code by He et al. (2022). During training, a random portion of SST patches is masked/removed, and the encoder only processes the unmasked

and included the http link in the acknowledgement section:

Acknowledgements. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004) and was funded by the Advanced Information Systems Technology (AIST) program. We acknowledge the independent but similar research effort conducted by Angelina Agabin and J. Xavier Prochaska et al. The MAESSTRO code is derived from https://github.com/facebookresearch/mae.

3. Line 8: “It has exceptional efficiency, requiring three orders of magnitude (a factor of 5000) less time.” Compared to what? “We compared the MAESSTRO with the interpolation methods used in the paper as our benchmarking. We have revised the manuscript accordingly.”

This is still not clear to me since most neural network implementations run on a GPU (or other accelerator). Have the "conventional approaches" been ported to the GPU or are you also comparing GPU vs CPU here?

The reviewer is right on the ambiguity of using CPU vs GPU. While MAESSTRO was trained on GPUs, we tested the inference only on CPU, so the comparison with CUP-based interpolation is still valid. We clarified this point in the text to be explicit of this comparison and interpretation. Now the sentence in the abstract is revised to

error (RMSE) of under 0.2°C for masking ratios of up to 80%. The application of the trained MAESSTRO has exceptional efficiency, requiring three orders of magnitude (a factor of 5000) less time compared to conventional approaches cubic radial-basis interpolation and Kriging used in this paper tested on a single CPU. The ability to reconstruct high-resolution SST fields under cloud cover has important implications for understanding and predicting global and regional climates, and detecting small-scale SST fronts that play a crucial role in the exchange of heat, carbon, and nutrients between the ocean surface and deeper layers. Our findings highlight the potential of deep learning models such as MAE to improve the accuracy and resolution of SST data at kilometer scales. It presents a promising avenue for future research in the field of small-scale ocean remote sensing analyses.

4. Line 175: “radial-basis bicubic interpolation” Can you give the equations of this interpolation method. Can it account for noise in the observations?

“The mathematical form of the Cubic RBF is given by: $\phi(r) = r^3$, where $\phi(r)$ is the cubic RBF, r is the radial distance from the center of the function where we would retrieve the interpolation. A linear combination of a set of these cubic RBF will yield a broad smooth function that can be best fit to scattered unstructured data as the cloud-covered SST images. We used the algorithm implemented in scipy.interpolation.RBFInterpolator. The smooth factor was set to 0 to perfectly fit the data at the available data point as we assumed zero noise, but it in principle can accommodate specified noise level by adjusting the smooth factor (whether the fitted surface goes through observations exactly or not). We added some description in the text.”

Please don’t give just the equation of a cubic function, but the equation of the interpolation method: the equation of how the field is obtained at any location given the data and the “smooth factor” (and any other parameter). If this description would be too long, please at least, reference an article describing this method.

Thanks for clarification. We added the following text to make it more explicit.

185 Given an image with a total of M pixels, where N pixels are missing, the missing pixels are filled using cubic-RBF interpolation. The value at each missing pixel x_j is estimated by:

$$T(x_j) = \sum_{i=1}^{M-N} c_i \phi(\|x_j - x_i\|) \quad \text{for} \quad j = 1, \ldots, N,$$

where c_i are the coefficients associated with the known pixels x_i, and $\phi(r) = r^3$ is the cubic radial basis function. These coefficients c_i are determined through an optimization process based on the known pixel values.

5. “In our validation datasets sourced from LLC2160, we have employed a consistent random masking strategy, where the masks were generated by the machine learning code automatically. Realistic cloud coverage was not incorporated in our first submission. We have now conducted additional tests that include cloud masks derived from VIIRS data. These masks were applied to the simulations to enhance the realism and applicability of our results. (Figure11)”

The _manuscript_ is still not clear about how the cloud mask was chosen. In particular, what mask is chosen for the RMSEs and correlations in Figure 9. The authors should also acknowledge that the RMSE for all methods is probably too low compared to the case where cloud masks have a larger spatial extent.

For the choice of masking, we added clarification in the text.

105 The absence of cloud cover in this dataset (i.e., complete SST visibility) enables its use as ground truth when evaluating the MAE’s SST reconstruction performance in masked regions. During the MAESSTRO training and validation, random patch masks were automatically generated (e.g., shown in the middle column in Figure 2). It is a question whether those random pixel masking represent real cloud shape. Even though we focus on simulations, we do illustrated a scenario with a real-cloud subtracted from satellite data and discussed in Section 5.

Some limitations have been acknowledged by the authors in the review, but they are not clearly stated in the manuscript. Please inform readers about these limitations in the manuscript (how parameters of the Kriging method was chosen, the mask cloud, training on model data).
We also acknowledge the low noise from the random masking and bigger continuous cloud would lead to larger errors. The discussion was included in the following paragraph in the discussion section.

310 The artificially-generated cloud mask employed in this study resembles spotted cloud patterns, such as those produced by altocumulus clouds with relatively uniform spatial distribution. Nonetheless, it is important to acknowledge that significant errors are anticipated in areas with continuous cloud masks over an extensive area as shown in Figure 11. This is due to the model’s inability to establish connections between data points across large spatial distances. Small-scale features may be lost if the gap created by clouds is too large. Future investigations could explore methods for integrating complementary information,
315 such as incorporating low-resolution, cloud-free microwave SST data and/or cloud-free sea surface height data from altimeters.

The manuscript was also not updated with some additional information given here, such as which Kriging variant was used, what happens if a single pixel is missing in a 4x4 patch,.... Please make sure that clarifications given here are also reflected in the manuscript. If it is already mentioned, thank you for providing the line numbers.

The Kriging method details are now included in section 3 as shown below.

175 **3 Evaluation on a sample SST tile from LLC2160**

We compare MAESSTRO's performance on the LLC2160 test set with Kriging method Matheron (1963), a commonly employed gridding technique for irregular geospatial analysis and the foundation of a widely used SST product (Reynolds et al., 2007), as well as cubic radial-basis function (cubic-RBF) interpolation, which is less popular but surprisingly outperformed Kriging methods in our case (details are in Section 3.2 and Table 2). Two variogram models were used in Kriging method,
180 linear and Gaussian. The linear variogram does not have a range but the 128x128 image size effectively limit the range of the variogram at 128 pixels. The linear variogram is dynamically calculated based on each provided masked SST fields. The Gaussian variogram has 0.1^oC, 20 pixels, 0.001 ^oC for the sill, range, nugget parameters, respectively. The cubic-RBF fits a cubic function for each data points with a basis function $\phi(r) = r^3$, where r represents the distance of a center point to a point with values. The cubic-RBF is featured with a smooth surface/features that is suitable to the turbulence nature shown by
185 the SST images. The LLC2160 SST tiles consist of 128x128 grid points (approximately 512x512 km ^{2} in the mid-latitudes). The grid spacing is twice as large as that in LLC4320, thereby allowing us to test MAESSTRO's ability to generalize across different spatial resolutions.

For missing pixels in a 4x4 patch, we added the following highlighted sentence.

80 An extensive hyperparameter search resulted in MAESSTRO using an MAE variant with a patch size of 4 and a ViT-Tiny encoder (Dosovitskiy et al., 2020). Here, "patch 4" refers to the process of dividing the input image into non-overlapping patches of size 4x4 pixels. The ViT model is designed to handle images as sequences of patches, much like a language model processes sequences of words or tokens. By breaking down the image into 4x4 pixel patches, the ViT encoder can effectively process and analyze the spatial structure and features of the image. This approach allows the ViT to leverage the advantages
85 of the transformer architecture in computer vision tasks. In this proof-of-concept study, we exclude patches if any of the 4x4 pixels contain missing values.

Thanks again for reviewing the manuscript.