Reviewer 1

In this manuscript, Koo et al. describe the application of several variants of a graph neural network to emulate ISSM at Helheim Glacier. I am unsure as to exactly what this emulator does- whether it is similar to IGM in producing an approximate solution operator to the (in this case coupled) momentum and mass conservation equations or whether it is more similar to He (2023) in being geometry specific- but in general terms it is trained to reproduce the predicted ice velocity and thickness as a function of time and space.

I think that this paper has the potential to be a useful contribution to the literature, and the goal of coming up with emulators that operate naturally in the same discrete setting as FEM-based ice sheet models is worthy. Additionally, the saliency with which the neural networks either learn or memorize (I'm not entirely sure which) the model's behavior is impressive. However, the manuscript needs significant clarification (and in some cases moderation) of its claims in order to assess their veracity and utility. In particular, the methods are unclear and not reproducible- as mentioned above, I am not clear what the features used for prediction actually are. Additionally, the paper does not include a fair representation of the computational costs of the proposed methods. Finally, the paper makes many claims that its proposed methodology is better than others without providing sufficient evidence to back up that claim. Detailed comments are below.

- Thank you for your constructive comments. We review your comments carefully and will make necessary changes accordingly.

L63 Downs (2023), which is already referenced in this paper, provides significant insight into this question, and also serves as another example of a surrogate model being used to infer calving dynamics at Helheim Glacier (though not a GNN).

- We will add brief explanations about how Downs et al. 2023 inferred the calving dynamics in Helheim Glacier, and how our approach differs (although we consider our application as an illustrative example).

L75 The acronym VM should be defined here.

- VM is already defined in L29.

L78 Physically, why should the stress threshold have to be calibrated on a glacier-by-glacier basis?

- This is a good point that is a bit out of the scope of this paper. Previous studies (e.g. Morlighem et al. 2019) found that most glaciers in Northwest Greenland had a threshold around 1 MPa (see the first paragraph of their results section). But they also found that some glaciers "needed" a different threshold, even though it is not fully clear why that is the case. Some of these variations could be due to a bias in the model initialization or the forcing, but also due to different levels of damage of individual glaciers, which is not necessarily captured by the model. We will add a short explanation to the text.

Sec. 2.3 This section should include some earlier literature. It would be worth looking at Tarasov et al. (2012, https://doi.org/10.1016/j.epsl.2011.09.010) and Brinkerhoff et al. (2021, https://doi.org/10.1017/jog.2020.112).

- We will add Tarasov et al. 2012 and Brinkerhoff et al. 2021 to Section 2.3.

L123 I don't understand why GNN's would be particularly suited for ice front migration relative to other architectures.

- We apologize for missing a detailed explanation. We meant that this was particularly attractive for finite element ice sheet models because of how they typically discretize the model domain. ISSM is a numerical ice sheet model that relies on unstructured meshes: it uses a finer resolution in the fast ice region and a coarser resolution in the slow ice region to optimize computational efficiency. However, CNNs inherently rely on a fixed resolution (regular grid) for all points, which may not capture the detailed resolution at the ice front. If we want a sufficiently fine resolution at the ice front, the grid size of the CNN would be very fine, which increases the computational demand exponentially. On the contrary, since GNNs directly use the ISSM unstructured meshes, they can keep the advantages of the ISSM numerical simulations in a more natural way. Hence, GNNs can be a better option to (1) obtain accurate ice front mitigation by embedding the interaction between neighboring nodes, (2) obtain a sufficiently fine resolution to capture ice front migration, (3) minimize the computational load. We will add a detailed explanation of why we chose GNN as the backbone architecture.

L146 'adjacent' → 'adjacency'.

- Done.

L149 Finite elements are, at their core, interpolants. Does bilinear interpolation here just mean that the FEM solution is evaluated at grid points? Or is some other interpolant introduced?

- Yes. Since the FEM solution is not provided as regular grid points, we calculate the velocity and thickness solutions for regular grid points from the FEM results.

L179 the square root does not need to be defined.

- Done.

L180 'of' → 'with'.

- Done.

L185-187 I don't think that this description makes sense. The architecture *for some particular hidden layer* weights different adjacent nodes differently for an operation that is otherwise the same as vanilla graph convolution. The resulting convolutions are then stacked– graph convolution is stacked, but the attention operation is internal to that.

- Correct. This self-attention operation is internal to a hidden layer. We will add some clarification about it.

Eq. 6: I am not sure whether this equation is correct, but I am also not sure whether it's necessary- it seems like maybe something very detailed that appears in the reference (although it does seem weird to concatenate the projected feature vectors like this, which would seem to make the attention scores sensitive to the order of arguments). Maybe okay to forego?

- We used the same equation from the reference paper (Velickovic, 2018), but we remove this equation from the revised manuscript because it is just a detailed version of Eq. 5.

Fig. 3 I don't think that this figure is helpful for illustrating how either of these models work.

- This figure was intended to illustrate the concept of attention and equivariance. We remove this figure from the revised manuscript.

Sec. 4.3 This section is really difficult to understand. One thing that I can glean from this is that this is operation is O(n2) in the number of graph nodes- does that have any implications for performance?

- We add more explanation about the EGCN model in Section 4.3. We cannot explain every detail of how the EGCN works, but the details about the EGCN, including how this architecture guarantees equivariance, can be found in Satorras et al., (2021).
- Technically speaking, EGCN is an operation of $O(n^2)$ because it operates on all graph nodes, while the GCN and GAT operate only on the adjacent nodes. That is, the EGCN operates on all graph nodes to preserve equivariance in the entire graph. Thus, the EGCN requires more processing time than the GCN and GAT, as shown in Section 5.2. We add more discussion about this in Section 5.2.

Sec. 4.5 This description of the model's inputs and outputs should appear at the beginning of the methods section. Furthermore, specifically what these features mean needs to be much more clearly described- as it stands, I cannot assess the quality of this work because, despite looking at both the manuscript and the linked code, I cannot tell what this emulator is building a mapping between. There are a few things implications to mention associated with this. First, it appears that the time t is explicitly included as a feature. This then implies that the surrogate is not time-invariant and the mapping should be thought of as a tool for downstream analysis (like He (2023) or Downs(2023)) rather than as learning the solution operator for Stokes' equations (or Stokes plus continuity since the present work claims to predict thickness as well). This is fine-such models can be very useful- but it mandates a change in language to reflect the fact that it is unlikely that this method can generalize to other locations or times. Second, the velocity components at some previous time (it's not clear whether this is from the model's previous time step or at the beginning of the simulation- this notation needs to be modified to be more clear) is used as a feature. This is an unfortunate choice because one thing that we know about ice physics is that the velocity is approximately diagnostic of the geometry- if you know the latter, you can predict the former. In the absence of that property, how can this model be started? Is it the case that ISSM has to be run first before this emulator can be applied? What is a 'forwarding process'? What is a graph 'structure'? Is this just the collection of node/edge features? This section is essential to understanding what is going on (more essential even than architecture choice), yet it's only two paragraphs long and does not have sufficient information to allow for reproducibility.

- Thank you for your detailed comment. We will add more descriptions about the input datasets at the beginning of the Method section. Regarding the mapping of input and output features, Fig. 2 helps to understand how this emulator works between input and output features.
- Regarding your first point about the time-invariant, we agree that our emulator is not time-invariant. There are a few reasons why we include time as an input feature. We would like to make our emulator provide the variables that regular measurements are available for. This facilitates the comparison between emulator output and real observations and finding good parameterizations. Therefore, we select velocity and ice front, whose observations are available in real-time, as the output features of our emulators. In our emulator, it was necessary to include time as an input feature to indicate the temporal evolution of ice thickness. If real-time ice thickness observations were available, we would include ice thickness as both input and output features, exclude time from the input features, and make our emulator 100 % time-invariant. Unfortunately, however, real-time ice thickness observations are not available. Thus, we could not use real-time "previous-time-step" ice thickness as an input feature to predict "next-time-step" ice thickness. Instead, we include time and initial ice thickness as input features so that the emulator can indirectly see the impacts of the temporal evolution of ice thickness at a specific time step. We will add this explanation about input feature selection to Section 4.5 and mention that our method can have limitations when generalized to other locations or times.
- In this study, we would like to find the parameterization that matches best real observations into numerical models. Therefore, we design our emulator to predict the "next-time-step" velocity from the previous-time-step velocity with certain parameterization settings, assuming that the parameterization changes the ice velocity. This parameterization includes the calving threshold (sigma_max) and the geometry of the ice sheet as well, which is included as the input feature of the emulator (i.e., bed elevation and initial ice thickness).

- Here, the forwarding process just indicates the determination of output from input features via a neural network. ‘Graph structure’ is the collection of nodes and edges. We will clarify the meaning of these terms in the revised manuscript.

Eqs. 11, 12, 13 These are all standard definitions that do not need to be included here.

- Although the Eqs. 11-13 are standard definitions, we would like to keep them for completeness because some readers may wonder how these metrics are calculated.

L269 ‘out of sample’ is perhaps a bit of an overstatement- the degree of correlation between neighboring σ values would very likely be quite high- as a check, it would be interesting to see what error is induced by comparing model predictions made using σmax = 0.8 to the withheld test set values for σmax = 0.75 (or something like that). I expect the metrics would be similar because there is little difference between such small variations in the parameter. A more useful test would be to train on just the two extremal values (0.7,1.1) and see if it can still interpolate well.

- We agree that the degree of correlation between neighboring σ values can be high. In the Appendix, we will add some additional experiments with neighboring σ values: comparing σmax = 0.8 and 0.7 to the withheld test set values for σmax = 0.75, and comparing σmax = 0.85 and 0.95 to σmax = 0.90.

Table 1 These metrics are all so inflated as to be useless. Is it possible to come up with relative metrics that use more significant digits? I also think it would be better to combine Tables 2 and 3 with Table 1- it is useful to think of model accuracy relative to model size and expense.

- We found that the R values are too inflated because we calculated this metric for the entire glacier domain. However, the significant differences are only near the ice front and ice stream, while the other regions with slow ice show insignificant differences. We will recalculate these metrics only near the ice front and ice stream.
- Although combining Tables 1, 2, and 3 would be useful to see the model accuracy relative to the model size and expense, we are concerned that the combined table is too distracting because it includes too much information. Instead, we will combine Tables 2 and 3, which will make a new Table 2, so that the readers can focus on the model fidelity in Table 1 and computational expenses in Table 2.

Sec. 5.1 A single study does not establish the superiority of GNNs over CNNs for tasks such as these- it could (and very likely is) the case that the current results are incidental (or cherry picked) and that different researchers could find different conclusions. Furthermore, with respect to efficiency, there are many tricks that can be performed on CNNs to make them faster that have no analogue for an unstructured mesh, none of which were presumably included in the present analysis. The style of NeurIPS or similar notwithstanding, it is generally unhelpful to try to establish the primacy of one method over another in this way, and I would encourage the authors to either undertake a much more controlled and systematic comparison between methods or to reframe this as less of a competition.

- This is a good point, but we still argue that the advantage of using GNN is that it can directly use the unstructured meshes of ISSM. The unstructured meshes of ISSM (and other finite-element ice sheet models) are characterized by variable resolutions, which allocate high resolution to fast ice region and low resolution to slow ice region. While GNNs can directly use the finite-element mesh data structures, CNNs cannot do it as they only operate on regular grids. Since ISSM uses unstructured meshes, using CNNs as emulators for ISSM can introduce two problems: (1) the CNN grid with fixed resolution can lose dynamical details in fast ice areas; (2) the CNN grid requires unnecessary computational demands in slow ice areas. We submitted another research paper to another journal that applies a similar GCN architecture to another glacier in Antarctica. In this paper, we include more detailed discussions about the advantages of using GCN over CNN in terms of model fidelity and computational efficiency with various spatial resolution conditions. The preprint version of our paper is available at this link: https://arxiv.org/abs/2402.05291. In the revised version, we will highlight the advantages of GNN over CNN, particularly for the ISSM ice sheet model with unstructured meshes.
- We agree that several tricks can be applied to CNNs. However, in this study, we would like to just compare the most basic architectures. The GCN, GAT, and EGCN architectures are also very basic architectures designed for non-regular graph structures: GCN uses convolutional operations on graphs; GAT adds attention operation to the graph convolutional layer; EGCN uses equivariance graph convolutional layers to preserve equivariance in graph structures. Given that there are numerous modifications of basic CNN architectures, handling all of them is beyond the scope of this research paper. Nevertheless, since ISSM operates on unstructured meshes, GNNs have fundamental advantages over CNNs for emulating ISSM. In this paper, we would like to highlight the fundamental limitations of CNN in handling unstructured mesh, especially in delineating the calving front. We argue that GNNs perform better than CNN, at least within the architectures that we have tested, but others can find CNN architectures that may do a better job with exhaustive searches and numerous tricks.

Sec. 5.2 It is frankly absurd to not include even a mention of the computational cost associated with increasing the training data, which- so far as I can tell- must be repeated for any new geometry or parameter or location. Ignoring this cost does of course lead to much more impressive speed-ups, but these are not real. I would expect this problem to become considerably more severe when trying to use this technique to emulate models that are a function of more than a single parameter- the curse of dimensionality still applies. Furthermore, the notion that a GNN will be more suitable than a CNN for higher resolution modelling is another strawman because it ignores the fact that generating the cost of generating the training data (which is presumably more expensive than any network training) is also going to scale proportionally with resolution. I would encourage the authors to revisit this entire section with a more sober perspective aimed at delivering a factual assessment of the present work's utility.

- We agree that the repeated application of our GNN emulators to another location or parameter would require more computational cost. In this study, our simulation is only focusing on the VM calving law that requires only σmax for parameterization; however, if we use another calving law that requires additional parameters, it would take much more time to collect the simulation data from various parameter settings. Such computational demands with multiple parameters and different locations can be inferred from our results. Additionally, we want to emphasize that the computational significance of deep learning emulators is that they can find the statistical relationships between input and output features and the optimal parameter settings once they are trained from the provided simulation data. We will add a brief mention of this problem in Section 5.2.
- Generating training data is not a big task for GNN. Since GNN directly uses the meshes of ISSM, there is no significantly time-consuming extra workload. The only thing we need to do to generate a training dataset for GNN is to define the graph structures with the ISSM meshes, elements, and adjacency matrix of nodes, and it takes only a few seconds in a local machine. On the contrary, this additional workload is applied to CNNs rather than GNNs because CNNs require to interpolate all the features from the unstructured mesh of ISSM into regular grids.

L351 This is only true if the model's behavior in response to variations in this new parameter is as well quantified as it is to the parameter considered in this work, which may not be true, or may not be tractable.

- Agreed. Since it is not clear how the other parameters, which are not considered in our emulator, have impacts on calving, we will remove this part in the revised version. Instead, we will just briefly mention the limitation of the VM calving model.

Sec. 6 Again, I strongly urge the authors not to try to cast their work in terms of 'superiority'- the present work does not provide sufficient evidence for such blanket statements, nor is it necessary.

- Please see our previous response about the "advantages" of using GNNs over CNNs in replicating ISSM, which uses unstructured meshes, in terms of model fidelity and computational efficiency. We will however tone them down following your comment.

Fig. 7 This needs a more descriptive caption- I am struggling to see what these figures are showing.

- Sorry for the too-short caption. We will add a more descriptive caption for this figure.

L370–371 Is this statement really necessary?

- We will remove this statement in the revised version.

References

- Choi, Y., Morlighem, M., Wood, M. & Bondzio, J. H. Comparison of four calving laws to model Greenland outlet glaciers. Cryosphere 12, 3735–3746, https://doi.org/10.5194/tc-12-3735-2018, 2018.
- Wilner, J. A., Morlighem, M. & Cheng, G. Evaluation of four calving laws for Antarctic ice shelves. Cryosphere 17, 4889–4901, 2023.
- Morlighem, M., J. Bondzio, H. Seroussi, E. Rignot, E. Larour, A. Humbert, and S. Rebuffi (2016), Modeling of Store Gletscher's calving dynamics, West Greenland, in response to ocean thermal forcing, Geophys. Res. Lett., 43, 2659–2666, doi:10.1002/2016GL067695.
- Petrovic, J.J. Review Mechanical properties of ice and snow. Journal of Materials Science 38, 1–6 (2003). https://doi.org/10.1023/A:1021134128038
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y.: Graph Attention Networks, https://doi.org/10.48550/arXiv.1710.10903, 2018
- Satorras, V.G., Hoogeboom, E. and Welling, M., 2021, July. E (n) equivariant graph neural networks. In International conference on machine learning (pp. 9323-9332). PMLR.