Response to Reviewer

August 8, 2024

Detailed Comments
Responses are marked in blue.

- p. 2: new training procedure 30% faster - compared to 2D or original 3D?

  We apologize for the lack of clarification on that part. The improvement in training speed is compared to the 2D model, run with the exact same training configuration (same learning rate, same initial seed, same local and global batch size). We have clarified this in the text with the following phrasing:

  ...we propose a new training procedure that increases the speed of convergence for the 2D-Transformer model (compared to the original 2D model) by 30% without adjusting any other hyperparameters, increasing the possible performance of the model given a fixed compute budget."

- p. 3: what was the number of compute nodes? were local SSDs used in some form? if not, is mentioning them relevant to comparable studies (I believe such hybrid setups would be very peculiar to use)?

  The number of nodes was not fixed but is given through the number of GPUs utilized, i.e. the global batch size. We apologize for not stating this more clearly. Each compute node is equipped with four A100 GPUs. As stated in section 2.4, *"The validation case was trained with a local batch size of two on 120 A100-40 NVIDIA GPUs. Unless otherwise specified, all ablated models were trained with a local batch size of six on 120 A100-40 NVIDIA GPUs".*

  Runs always utilized all available GPUs on a node, splitting batches across them as evenly as possible. Hence, the number of nodes utilized for ablated models can be derived from the global batch size, divided by the local batch size (for ablation models, = 6) , divided by four (number of GPUs per node). The term "Ablated models" here comprises the models presented in Section 3.2 (Ablation study), but not the mini-batch experiments in Section 3.3 (Minibatch size effects). The local batch size of the minibatch study can be found in Table 3.

  The local SSDs were not used in this study. We agree with the reviewer that it would be interesting to investigate using them for optimized data I/O (Data staging from Lustre to local SSDs, binding all local SSDs using Beyond to have true shuffling between epochs, and then loading from there); this could also yield significant speed-up. However, such an implementation requires a substantial amount of platform-specific adaptation and would not

be translatable to other systems, hence reducing comparability for other users. We have thus removed the statement about on the local SSDs from the description of the compute environment.

- p. 5, fig. 1b: These plots do not show perfect sin/cos functions, they are skewed; reading the explanation in 2.6, I don't understand why. I believe this comes from tweaking them to match the original weight sums, but then I'm missing an explanation for this particular tweak. Also, they are not in phase as explained (l. 134); e.g., the maxima of U/V are slightly shifted (epoch 200 in (a), epoch 185 in (b)).

  We thank the reviewer for their observation and apologize for the confusion. The initial weights are distributed according to normal cosine functions. We also design the cosine distributions such that the velocity fields in the surface fields peak at the same time as in the pressure levels. However, at every epoch, we need to normalize the sum of the weights to match those of the original weight sums.

  The reason that the peaks are slightly offset is due to this weight normalization. Taking the U/V in plot (b) for instance: according to the regular cosine scheduling, the peaks should be at epochs 0, 100, and 200. However, since the weight of MSP at these points has a higher proportion of the overall weight, the relative weight of U/V is decreased. At epoch 85, the weights of MSP and T2M are very low, meaning that the U/V scales to a greater value.

  We have included a more explicit formulation of the weight distribution in the manuscript as follows:

  "Given an individual variable i at epoch e, for a period of n_{epochs} (in this case, $n_{\text{epoch}} = 100$). For the pressure variables Z, Q, T, U, V, $i = 1, 3, 2, 0, 0$, respectively. For the surface variables MSLP, U10, V10, T2M, $i = 1, 0, 0, 2$, respectively.

$$w(i, e) = \cos\left(\frac{2\pi}{n_{\text{epoch}}}\left(e - \frac{i}{2}\right)\right) + 1.1 \tag{1}$$

  At each epoch, the pressure variables are normalized to sum up to the original weights of $3.00 + 0.60 + 1.50 + 0.77 + 0.54$.

$$W(i, e) = w(i, e) \cdot \frac{3.00 + 0.60 + 1.50 + 0.77 + 0.54}{\sum_i w(i, e)} \tag{2}$$

  A similar procedure is applied to the surface variables, where the numerator is replaced with $1.5 + 0.77 + 0.66 + 3.0$.

  The ordering of the weights was designed such that the weights of the velocity and temperature variables (U, U10; V, V10; T, T2M) would peak at similar epochs."

- p. 8, l. 172: If I read figure 6 correctly, PGW-Lite failed to converge for sizes of 16, 32, and 480. For 64, it converged (hard to read the figure here but I believe there's a dashed red line just behind the solid red/orange line). This is in contradiction to what is written in the text.

  Thank you for your correction; we apologize for this error and we have updated this in the text:

"In contrast, the PGW-Lite model converges at sub-optimal loss values for minibatch sizes of 16, 32, and 480."

We have also split up the figure into two subfigures and changed the color scheme consistently across the manuscript to make the plots more legible.

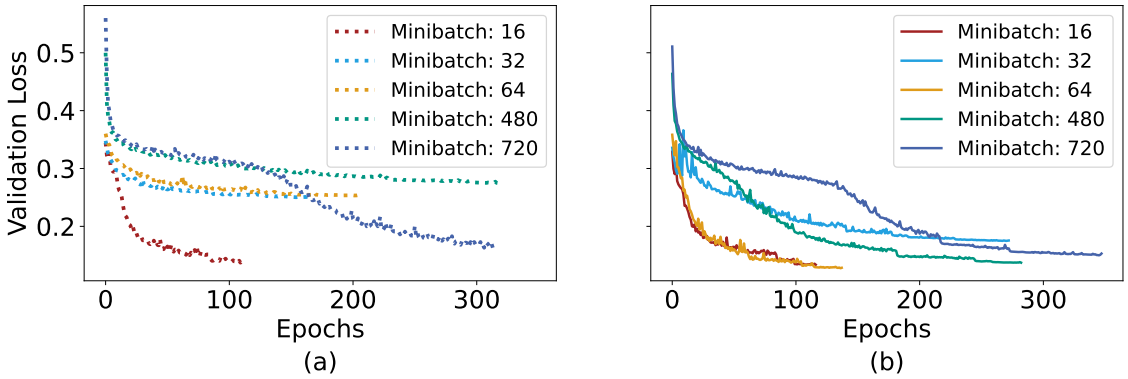

Figure 6. Validation loss of 2D-Attention and PGW-Lite as a function of epochs for global minibatch sizes of 16, 32, 64, 480, and 720.

- p. 8: The point of the minibatch study appears to be to 1.) analyze how minibatch sizes affect attainable loss/convergence and 2.) make a direct comparison on this between the 2D and 3D transformer approaches. For me as reader it would have been good to point this out here (it only became clear when reading discussion and conclusion) because it affects how one reads the text and plot.

  Thank you for the observation—we have introduced a new subsection (Section 2.6) in the methodology that motivates and explains the minibatch study:

  "A study directly comparing the PGW-Lite model, which features a 3D-Attention mechanism, and the 2D-Attention model was performed. Each of the two models was trained from scratch five times, varying the global batch size from 16, 32, 64, 480, 720. More experiments could not be conducted due to computational constraints. The details of the local and global minibatch size, as well as the average total GPU-hours/epoch and wall time per epoch, are shown in Table 3. All models were initialized with the same random seed, so the order of the training samples loaded from the data loader are identical."

- p. 8: Would different random seeds have a significant effect on convergence/attainable loss?

  We thank the reviewer for this question. Given the irregularity in the final loss values for the PGW-Lite model, we hypothesize that the use of different random seeds can have a significant effect on the attainable loss. Due to computational constraints, we did not perform multiple repetitions to study the effect of different seeds on the final loss values of these models. To address this, we have added the following explanation to Section 3.3:

  "In contrast, the PGW-Lite models converge at sub-optimal loss values for minibatch sizes of 16, 32, and 480. In these cases, the models were allowed to train until the learning rate dropped

to $3 \cdot 10^{-5}$. The irregularity of this behavior can be attributed to the model's sensitivity to the initial random seed."

- in general, while zooming helps, the colour scheme (use of yellow) and size of figures 6 and 7 makes them hard to read, particularly given that many lines relevant to discussion overlap.

  Thank you for your feedback. We have increased the size of all figures, particularly the ones with subfigures. We have also changed the color scheme to remove the yellow. As stated above, we have split up Figure 6 into two subfigures for clarity. We have also reduced the y-axis range on Figure 7 to show the plot more clearly.

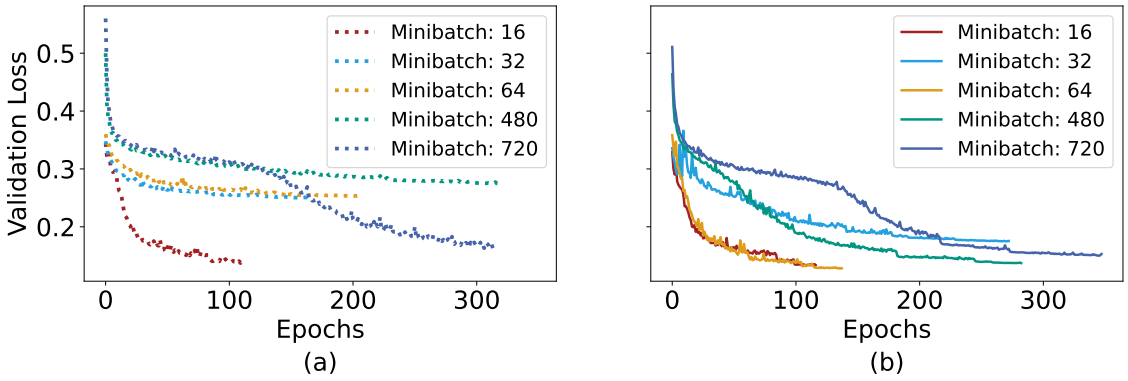

Figure 6. Validation loss as a function of epochs for global minibatch sizes of 16, 32, 64, 480, and 720 of (a): 2D-Attention and (b): PGW-Lite.

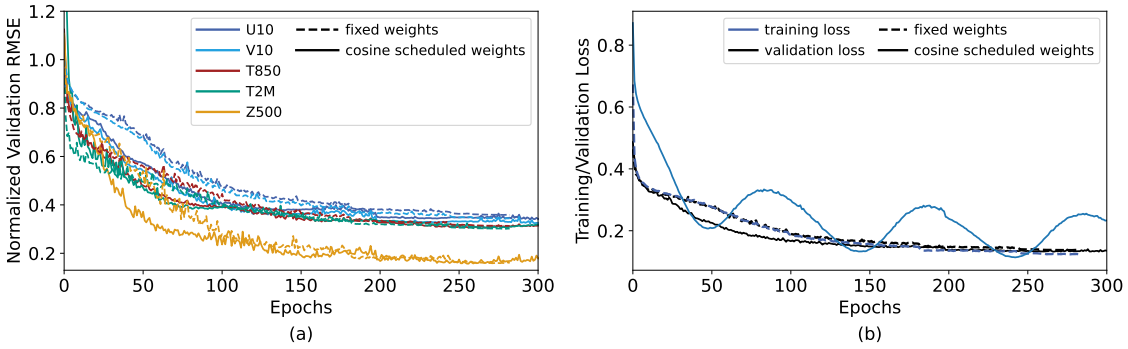

Figure 7. Error metrics for variable-specific weighted cosine loss scheduling applied to the 2D-Attention. The 2D-Attention model with PyTorch's ReduceLROnPlateau scheduler was maintained as the reference model and compared to the variable-specific weighted cosine loss scheduling model. (a): RMSE values of the 10m-U- and V-velocity, as evaluated on the validation dataset, as a function of the epochs. (b): training and validation losses for the two models.