Supplemental Material

Artificial Intelligence for the Earth Systems
Postprocessing of Ensemble Weather Forecasts Using Permutation-Invariant Neural Networks
https://doi.org/10.1175/AIES-D-23-0070.1

Postprocessing of Ensemble Weather Forecasts Using Permutation-invariant Neural Networks: Supplementary Material

Kevin Höhlein,[a] Benedikt Schulz,[b] Rüdiger Westermann,[a] and Sebastian Lerch[b,c]

[a] *Technical University of Munich*
[b] *Karlsruhe Institute of Technology*
[c] *Heidelberg Institute for Theoretical Studies*

1. Hyperparameter selection

Appropriate tuning of the training- and architecture-related hyperparameters of the respective model classes is essential to achieve a fair comparison. In what follows, we detail the hyperparameter settings chosen for the respective model classes, as well as the methods that led to the decision. To find good sets of hyperparameters for all model variants, we follow the suggestions by Godbole et al. (2023) and conduct an multi-phase parameter search, consisting of randomized parameter space exploration, followed by automated tuning of the hyperparameters using Bayesian optimization, and ablations to avoid excessive complexity of the models.

a. Model classes

As shown in Fig. 1, we impose a hierarchical classification on the model types for parameter tuning. On the highest level, we distinguish ensemble-based from summary-based models, which is the most severe differentiation, since models of both groups are trained on data with different information content. On the second level, we group the models by the architecture. This is relevant mainly for ensemble-based models, where encoder-decoder models process information differently from transformers. Following Schulz and Lerch (2022), all summary-based models use simple MLPs. The third level distinguishes with respect to posterior parameterization, i.e., DRN-type vs. BQN-type output parameterization. Note that all DRN models parameterize a truncated logistic distribution, both for wind-gust and temperature postprocessing. The fourth level is relevant only for ensemble-based encoder-decoder models and separates the models by the merging strategy used between member-wise encoder and decoder.

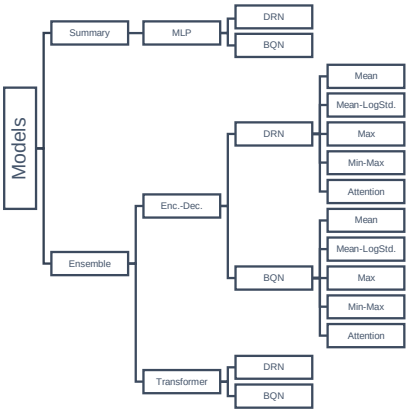

Fig. 1. Model hierarchy for parameter tuning. Models are grouped by 1) input type, 2) architecture, 3) posterior parameterization and 4) merging strategy (encoder-decoder models only).

b. Initial exploration

For each model class, we execute a number of initial trial runs to gain an understanding of how different parameters affect the model performance. Performance was assessed by comparing average losses on the training and validation parts of the respective datasets. We found that the batch size for training can be chosen flexibly, as long as the remaining parameters are tuned accordingly. We saw that similar sets of training-related hyperparameters (learning rate, patience for early stopping, dropout rates) lead to different results 1) when predicting different lead times and 2) when changing architecture-related hyperparameters (number of layers, channels per layer) of the respective model classes. Especially encoder-decoder and transformer models with larger MLP components appeared to profit from dropout-based regularization. We also found that the dimension of the station embedding affects the performance of ensemble models. Starting from a default value of 10 (cf. Schulz and Lerch 2022), trials showed that smaller values improve the quality of single-model predic-

Corresponding author: Kevin Höhlein, kevin.hoehlein@tum.de

tions but penalize the accuracy of the ensemble prediction, while larger values lead to overall reduced performance. The station embedding was therefore not considered in the subsequent parameter search. Similarly, the model depth was excluded early on as a hyper parameter since there was no indication that increasing the model depth to more than three layers (three attention blocks for set transformers) leads to better results.

c. Bayesian parameter search

Based on the findings of the parameter space exploration, we designed Bayesian optimization experiments for the ensemble-based model classes. The complete set of hyperparameters is split into training-related (cf. Tab. 1) hyperparameters and architecture-related hyperparameters (cf. Tab. 2). Bayesian search is used for the former, grid search was found more reliable for the latter. During Bayesian tuning, we use the Bayesian optimization functionality of Ax (`https://github.com/facebook/Ax`) and optimize for a multi-objective, consisting of validation loss and training time. Due to time constraints, we train only one model per parameter configuration, i.e., ensembling multiple models is not considered. Once an optimal setting was identified, a full ensemble of 20 models is trained and the best configuration of architecture-related parameters wrt. validation scores is selected. To save additional time in tuning encoder-decoder models, we choose to tune parameters only for the attention-based merger, which was found to yield the best results in the early exploration phase. Other merger configurations are considered in ablation studies. The respective parameter selections and ranges for parameter search are summarized in Tabs. 1 and 2. Hyperparameters are tuned separately for different datasets and lead times, since models for larger lead times were found to require stronger regularization. The final hyperparameters are shown in Table B1 in Appendix B of the main paper.

Table 1. Search space of training-related hyperparameters.

Model class	Parameter	Range
DRN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	$[12, 24]$
BQN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	$[12, 24]$
ED-DRN	learning rate	$[10^{-5}, 10^{-2}]$
	patience	$[12, 24]$
	encoder dropout rate	$[5\times10^{-3}, 0.5]$
	decoder dropout rate	$[5\times10^{-3}, 0.5]$
ED-BQN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	$[12, 24]$
	encoder dropout rate	$[5\times10^{-3}, 0.5]$
	decoder dropout rate	$[5\times10^{-3}, 0.5]$
ST-DRN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	$[8, 24]$
	transformer dropout rate	$[5\times10^{-3}, 0.5]$
	decoder dropout rate	$[5\times10^{-3}, 0.5]$
ST-BQN	learning rate	$[10^{-5}, 10^{-3}]$
	patience	$[8, 24]$
	transformer dropout rate	$[5\times10^{-3}, 0.5]$
	decoder dropout rate	$[5\times10^{-3}, 0.5]$

Table 2. Selection of architecture-related hyperparameters.

Model class	Parameter	Values
DRN	channels (first layer)	32, 48, 64
BQN	channels (first layer)	32, 48, 64
	polynomial degree	8, 12, 16
ED-DRN	channels (encoder)	48, 64
	channels (decoder)	48, 64
ED-BQN	channels (encoder)	48, 64
	channels (decoder)	48, 64
	polynomial degree	8, 12, 16
ST-DRN	channels (transformer)	48, 64
	channels (decoder)	48, 64
ST-BQN	channels (transformer)	48, 64
	channels (decoder)	48, 64
	polynomial degree	8, 12, 16

d. Ablations

To evaluate the effect of various changes in model architecture, we conduct ablation experiments, in which we disable specific aspects of the training or replace them by other mechanisms. We conduct the following ablations:

No dropout: For ensemble-based models, the Bayesian parameter search indicated that randomized dropout during training can improve model performance. To evaluate the veracity of this finding for the full ensemble model, we retrain an ensemble of 20 models with optimal Bayesian hyperparameters but with dropout disabled, and evaluate the performance of 10-member average models. The results show that dropout slightly worsens CRPS as well as PI length for DRN-type ED models. For the final comparison, dropout is therefore disabled in this class.

1D vs. 2D yearday embedding: In contrast to the original DRN and BQN Schulz and Lerch (2022), ensemble-based models in this study use a 2D embedding of the day of year information through sine in cosine modes, which was found profitable during the initial exploration. We evaluate the effect of using

both variants but do not find significant differences upon closer inspection.

Spherical vs. plain lat-lon embedding: In contrast to the original DRN and BQN Schulz et al. (2021), ensemble-based models in this study use a 3D embedding of station positions in terms of spherical coordinates. The 3D embedding offers a more accurate representation of the spherical geometry of the Earth, which may have an advantage when considering stations all around the globe. We ablate this design choice and supply the models with plain latitude- and longitude coordinates instead (whitened and normalized). For the data considered in this study, the location embedding is found to be of minor importance. We attribute this to the fact that for both datasets all stations are located in Europe, such that potential distortions due to coordinate projections are sufficiently small to not affect model performance.

Choice of the merger: We compare the model performance of ensemble-based encoder-decoder models with different merging algorithms. Attention-based merging is overall favorable for both DRN- and BQN-type models. For DRN, the results for long lead times suggest that the training outcomes suffer from instability and don't always converge to good local optima. BQNs are less prone to this behaviour.

2. Additional results

a. Comparison of PI length for DRN and BQN

Here, we briefly comment on the differences arising from the underlying distribution type, which agree for both data sets considered in this work. Plotting the PI length on the nominal level for different choices thereof in Fig. 2, we find that the choice of the forecast distribution defines the magnitude of the PI length. Up to a nominal level of around 90%, the PI lengths of the DRN and BQN forecasts both increase linearly at the same length, but for higher levels, they start to deviate as the PI lengths of the DRN models increase exponentially, while that of the BQN models still increase linearly. This can be explained by the fact that the BQN forecast distribution has compact support defined by the coefficients, whereas the distributions of DRN models are heavy-tailed with support on the positive real line. However, a detailed comparison of these two approaches for modeling tails of distributions in terms of predictive accuracy and appropriateness is not discussed in the following.

The data sets we consider include ensembles of three different sizes with ensemble ranges that correspond to PIs at the 83.33%, 90.48%, and 96.15% levels. Comparing the BQN and DRN forecasts for these different levels, we find that for 11-member EUPPBench reforecasts the DRN

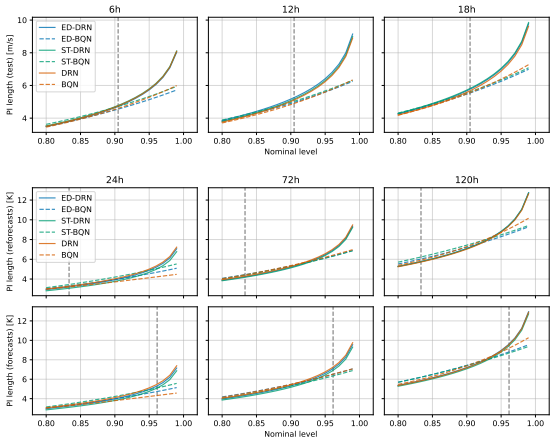

FIG. 2. PI length as a function of the nominal level for 11-member reforecast and 51-member forecast ensembles. PI lengths are computed based on averaged predictions of 10-member model ensembles. The shown values are averaged over 50 such ensemble models. The vertical dashed lines indicate the nominal level of the underlying ensemble.

models result in sharper forecasts, while the BQN models generate sharper forecasts for the 51-member ensemble forecasts. In the case of the wind gust data, the 20-member ensemble corresponds to a nominal level close to the intersection of the PI length curves, hence the differences between the distribution types are less pronounced than for the EUPPBench data.

Fig. 3 shows the relative deviation of the PI lengths of the permutation-invariant models from the mean-based models. In general, we cannot conclude that either of the approaches produces sharper forecasts, in general. Comparing the behavior over the nominal levels, we find that differences in the PI lengths are constant for the DRN models, while we see a trend in the differences for the BQN models, namely, either a monotonic increase or a decrease. This means that the permutation-invariant BQN models lead to different behavior in the tails of the distribution. In the case of wind-gust postprocessing, we see a decrease in the deviation, which corresponds to a lighter tail. For both temperature datasets, we see an increase in deviation for 24h lead time, i.e., a heavier tail, whereas for 120h lead time a decrease is seen as in the wind-gust case. At lead time 72h, the difference remains roughly constant with values close to zero on reforecast data, indicating approximately equal weighting of the tails. On the forecast dataset, a slight negative trend is observed. Further, the confidence bands, corresponding to the 95% interval of the mean, show that the tendency is fixed for each of the nine different forecast cases.

b. Variability over neural network resamples

Comparing the differences between the permutation-invariant model classes for the wind gust data, we find

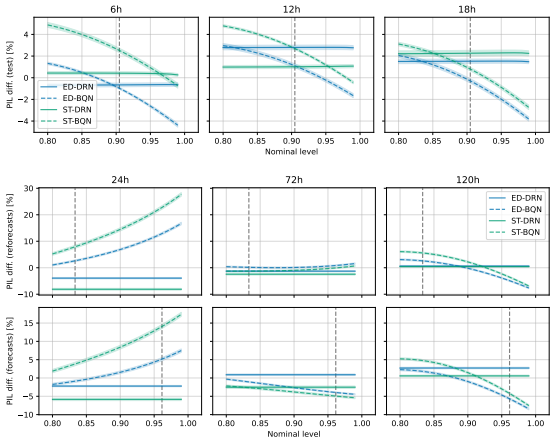

FIG. 3. Relative deviation of PI length from the mean-based model as a function of the nominal level for 11-member reforecast and 51-member forecast ensembles. PI lengths are computed based on averaged predictions of 10-member model ensembles. The shown values are averaged over 50 such ensemble models. PI lengths for DRN and BQN are considered as the baseline for the respective permutation-invariant models. The vertical dashed lines indicate the nominal level of the underlying ensemble.

only minor differences. Fig. 4 explores these differences in more detail. Shown are boxplots of the CRPS and PI length of the permutation-invariant network classes distinguished by the employed ensemble merging algorithm. For all lead times and network classes, the same pattern is observed. Mergers based on extreme values such as the minimum or maximum result in a worse CRPS and wider PIs than those based on calculations of the mean and attention models. The same pattern as for the extreme value models is observed for the set transformer models, that is, a worse CRPS and wider PIs. With respect to the benchmark models, we find that the CRPS is larger for almost all cases considered. In terms of the PI length, we find that the mergers based on the mean mostly result in slightly smaller PIs. The overall differences observed are however only on a small scale. Attention-based merging is employed for all experiments in the main text.

c. *DRN for EUPP: Truncated logistic vs. truncated normal posterior*

To maintain close similarity between the wind gust and the EUPPBench case study, we construct DRN models that parameterize a truncated logistic posterior distribution, analogous to the case of wind gust postprocessing. Note that temperature observations T_{obs} are recorded in Kelvin, such that $T_{\text{obs}} > 0$ holds, which justifies the truncated logistic distribution as a valid choice for temperature postprocessing. However, a more common choice would be the (truncated) normal distribution (e.g., Gneiting et al. 2005; Rasp and Lerch 2018). To ensure a fair comparison, we validate here that the design choice of using the truncated logistic posterior does not affect the postprocessing capabilities of DRN negatively.

Tab. 3 displays the prediction metrics as obtained for DRN models with truncated logistic (DRN-TL) and normal posterior (DRN-TN). Note that DRN-TN has undergone the same hyperparameter search as DRN-TL models and that the model selection was based on validation scores, not on the test scores that are shown in Tab. 3. DRN-TN models are identical in architecture to the DRN-TL counterparts, including the softmax constraint for both the location and the scale parameters, but are trained to optimize an analytical expression of the CRPS for a normal distribution Gneiting et al. (2005), given the training data. For evaluation, we compute the CRPS based on a zero-truncated normal distribution. The distinction is made to avoid numerical instabilities that are caused by the truncation terms during training. Due to the magnitude of the temperature observations and the expected value range of the fitted distributions, $T_{\text{obs}} \sim 300\,\text{K}$, $\Delta T_{\text{obs}} \sim \pm 30\,\text{K}$, however, this does not have a large effect on the final outcome. As seen in Tab. 3, we find that DRN-TL and DRN-TN score roughly identically in terms of CRPS. Due to the heavier tails, the PI length of DRN-TL models is slightly larger, and the coverage probabilities are met slightly more accurately for DRN-TN. Fig. 5 displays calibration histograms for both model variants. We note that both models overestimate high-temperature extremes slightly at lead times 24h and 72h on reforecast data, and underestimate low-temperature extremes at lead times 72h and 120h. Similar findings apply to the case of forecast data. Despite minor differences, both forecasts appear overall well-calibrated and we do not see reasons to expect qualitatively different results in our study when exchanging the truncated logistic posterior with the truncated normal.

d. *BQN: Full ensemble vs. summary statistics as predictors*

To enable a more direct comparison between DRN- and BQN-type models, we deviate from prior work and train MLP-based models with BQN posterior using the mean and the standard deviation of the primary predictor ensemble (t2m for EUPPBench, VMAX-10M for wind gusts). This differs from the work by Bremnes (2020) and Schulz and Lerch (2022), who use the full ensemble in sorted order as input to the BQN models. Here we validate this design decision and show that both approaches yield qualitatively similar results. The case of EUPPBench is of particular interest here, since ensemble-based models have to cope with different ensemble sizes in the reforecast and forecast test cases.

But first, we compare the forecasts for the wind gust data set. Table 4 shows the evaluation metrics for the different lead times, where we observe almost identical

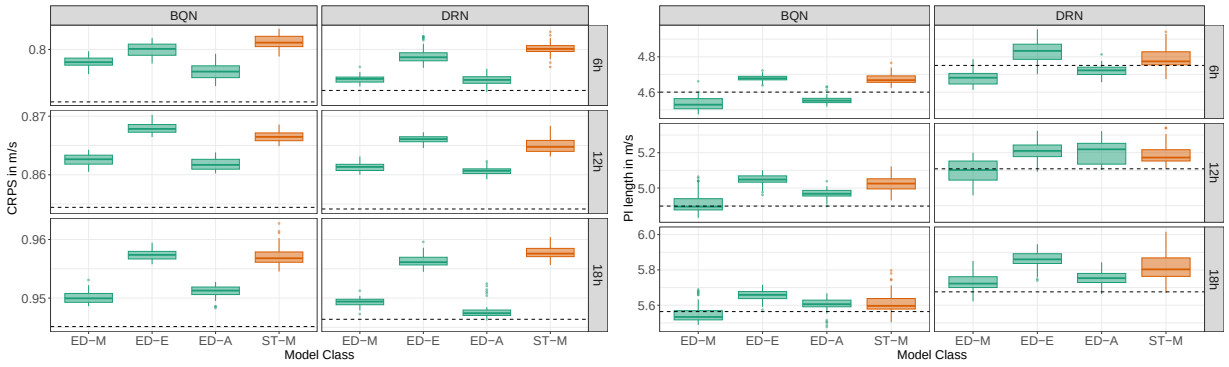

FIG. 4. Boxplot of the CRPS (left) and PI length (right) in m/s for the different lead times and model classes, the dashed lines display the value of the corresponding benchmark. The last letter refers to merger, where M comprises the mergers associated with the mean (mean, mean-logstd, weighted-mean, weighted-mean-logstd), E the extremes (max, min-max), and A the attention models. The boxes are calculated based on the ensemble resamples, where for each repetition we picked that configuration within the model class that yields the smallest CRPS on the validation set.

Lead Time		24h			72h			120h		
Dataset	Method	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.
Reforecasts	DRN-TN	0.67	3.27	83.87	0.86	4.14	83.41	1.20	5.93	84.17
	DRN-TL	0.67	3.28	84.16	0.86	4.27	84.58	1.19	5.70	83.09
Forecasts	DRN-TN	0.64	5.01	96.91	0.81	6.58	97.35	1.14	8.62	96.83
	DRN-TL	0.64	5.48	97.92	0.80	7.21	98.37	1.13	9.58	98.28

TABLE 3. Test scores for DRN architectures with truncated normal (TN) and truncated logistic (TL) posterior on EUPPBench data for different lead times. PI length and coverage are computed for a significance level corresponding to an 11-member ensemble (~ 83.33%) for reforecasts and a 51-member ensemble (~ 96.15%) for forecasts.

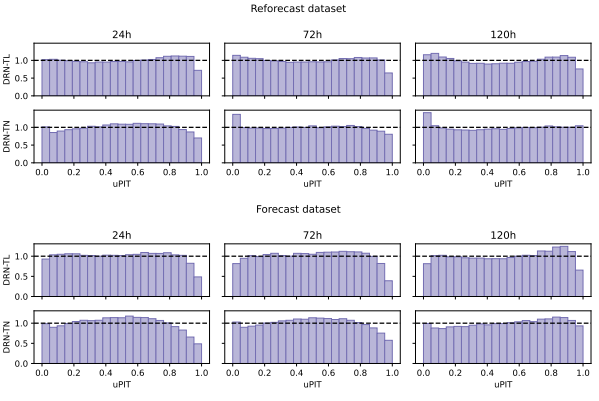

FIG. 5. Calibration of postprocessing models on 11-member EUPP-Bench reforecast ensembles (top) and 51-member forecast ensembles (bottom).

results. The negligible differences seen only in the PI lengths and coverages might also be a result of the different realizations of the underlying network ensembles trained. For the wind gust data, we conclude that there are no differences in the predictive performance between the two variants.

Tab. 4 shows the model scores on EUPPBench reforecast and forecast datasets, comparing a BQN model informed with summary statistics (mean and standard deviation, BQN-Sum) and the full ensemble model (BQN-Ens). Both models adhere to the hyperparameter configuration listed in Appendix 1. Most notably, BQN-Ens comes with a slightly larger PI length on reforecast data at 24h lead time, while yielding the same CRPS as BQN-Sum. According to PI coverage, BQN-Sum matches the theoretical value of 83.33% more accurately. For the remaining lead times, the differences are negligible. To make BQN-Ens applicable to the forecast dataset, which comprises more members per ensemble, we distinguish randomized subsampling (BQN-Ens-R, 11 out of 51 members, sampled without replacement) and quantile-based subsampling (BQN-Ens-Q). For the latter, we sort the 51-member ensemble in ascending order and pick the members with rank $(51+1)/(11+1)*i$, for $i = 1,...,11$, as the predictor ensemble. BQN-Sum yields marginally sharper forecasts at 24h and 120h lead time. The differences between the subsampling variants are negligible. Fig. 6 displays calibration histograms for all model variants. All histograms exhibit a wave-like structure, but appear otherwise very similar, despite slight differences in the placement of the distribution peaks. The

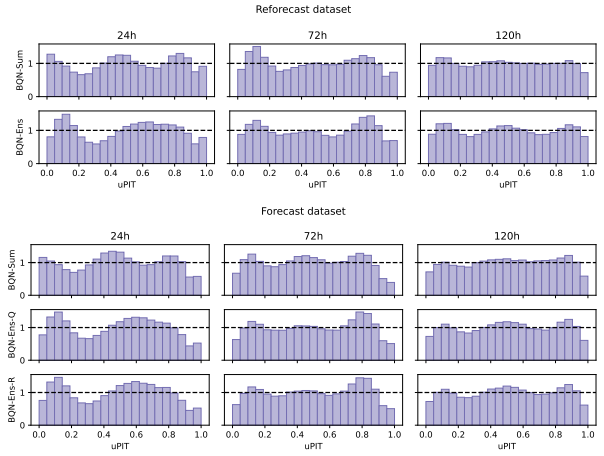

FIG. 6. Calibration of postprocessing models on 11-member EUPPBench reforecast ensembles (top) and 51-member forecast ensembles (bottom).

120h case achieves the most uniform calibration, overall. Again, hardly any differences are seen between the subsampling variants. We conclude that it is well justified to replace the full ensemble with the summary-based predictors for BQN models

3. Additional figures

Here we include figures, which are obtained using the methods in the main paper but are too exhaustive to include in the main text. We add an overview of the complete set of permutation feature importance values, shown in Fig. 7. We also provide illustrations of the ensemble-oriented permutation feature importance for all variables and lead times. The data for wind gust postprocessing are shown in Figs. 8 to 16, the data for the EUPPBench case are shown in Figs. 17 to 20. Note that for some predictors the bar charts show large variations. The reason for these behaviors is extreme outliers, which distort the statistics. However, such extreme cases are only observed for parameters of limited permutation importance (cf. Fig. 3 in the main text).

References

Bremnes, J. B., 2020: Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. *Monthly Weather Review*, **148 (1)**, 403–414, https://doi.org/10.1175/mwr-d-19-0227.1.

Gneiting, T., A. E. Raftery, A. H. Westveld, and T. Goldman, 2005: Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, **133 (5)**, 1098–1118, https://doi.org/10.1175/MWR2904.1.

Godbole, V., G. E. Dahl, J. Gilmer, C. J. Shallue, and Z. Nado, 2023: Deep learning tuning playbook. URL http://github.com/google-research/tuning_playbook, version 1.0.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, **146 (11)**, 3885–3900, https://doi.org/10.1175/MWR-D-18-0187.1.

Schulz, B., M. E. Ayari, S. Lerch, and S. Baran, 2021: Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting. *Solar Energy*, **220**, 1016–1031, https://doi.org/10.1016/j.solener.2021.03.023.

Schulz, B., and S. Lerch, 2022: Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison. *Monthly Weather Review*, **150 (1)**, 235–257, https://doi.org/10.1175/mwr-d-21-0150.1.

Lead Time		6h resp. 24h			12h resp. 72h			18h resp. 120h		
Dataset	Method	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.	CRPS	PI length	PI cov.
Wind gusts	BQN-Sum	0.79	4.60	90.23	0.85	4.90	89.65	0.95	5.56	90.70
	BQN-Ens	0.79	4.61	90.20	0.85	4.94	89.91	0.95	5.56	90.65
Reforecasts	BQN-Sum	0.68	3.19	82.90	0.87	4.43	85.87	1.19	5.91	84.75
	BQN-Ens	0.68	3.37	84.92	0.87	4.43	85.93	1.19	5.90	84.58
Forecasts	BQN-Sum	0.64	4.32	94.13	0.80	6.52	97.23	1.13	9.18	97.58
	BQN-Ens-Q	0.65	4.97	96.36	0.80	6.50	97.11	1.13	9.31	97.65
	BQN-Ens-R	0.65	5.00	96.39	0.81	6.56	97.11	1.13	9.30	97.65

TABLE 4. Test scores for BQN architectures with predictors based on the full ensemble (BQN-Ens) of primary predictors and predictors based on summary statistics (mean and standard deviation, BQN-Sum). PI length and coverage are computed for a significance level corresponding to a 20-member ensemble ($\sim 90.48\%$) for the wind gust data, an 11-member ensemble ($\sim 83.33\%$) for the EUPPBench reforecasts, and a 51-member ensemble ($\sim 96.15\%$) for the EUPPBench forecasts. In the case of the EUPPBench forecast data with ensemble-valued predictors, the full 51-member ensemble is subsampled randomly (BQN-Ens-R) or based on quantiles (BQN-Ens-Q) to match the 11-member training dataset.

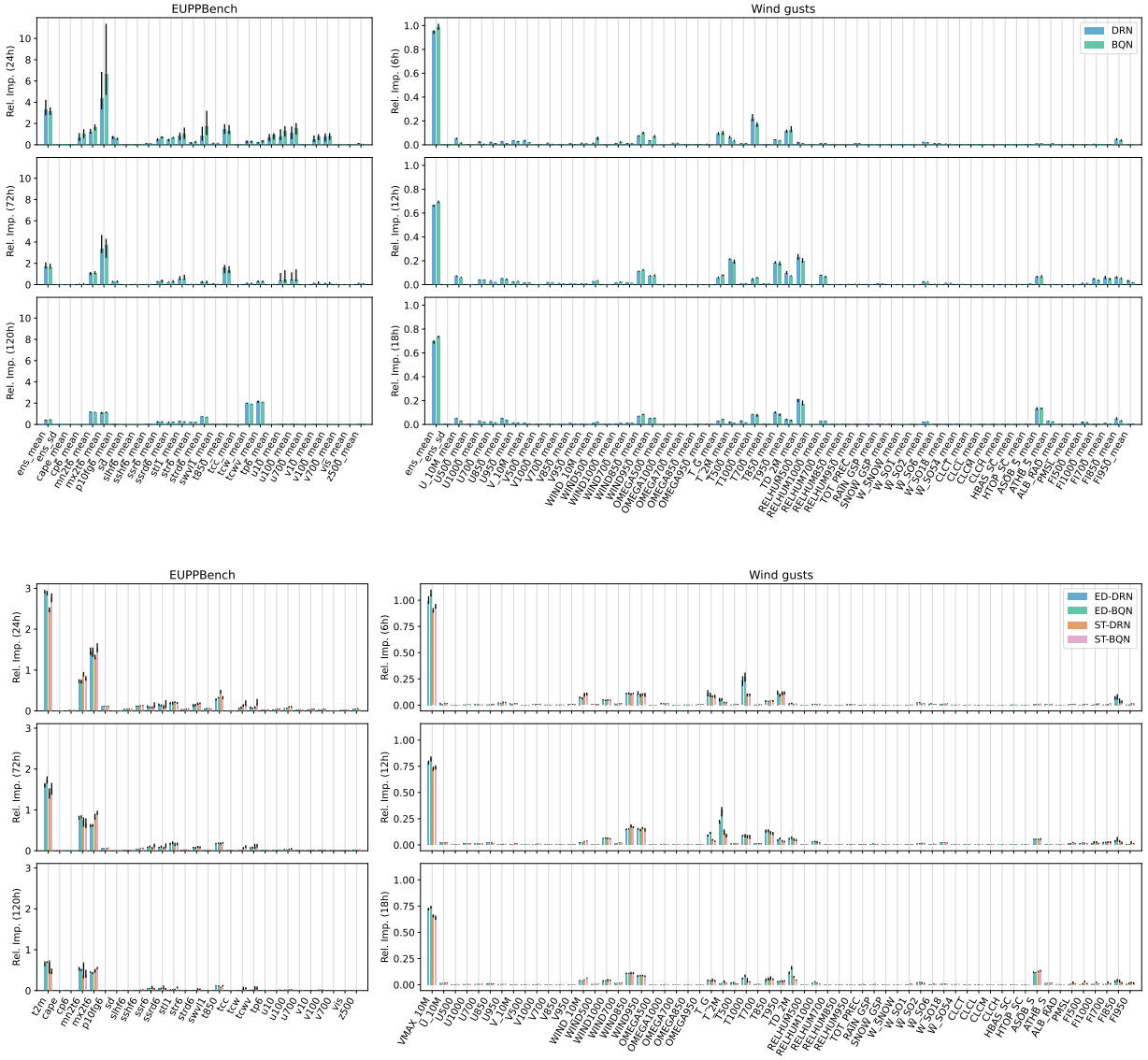

FIG. 7. Permutation feature importance for summary-based networks (top) and permutation-invariant models (bottom) for EUPPBench and wind gust postprocessing. Predictors named ens in the top figure correspond to the primary predictors t2m and VMAX-10M, respectively. The suffix sd indicates the ensemble standard deviation of the predictor. Same as Fig. 3 in the main text.

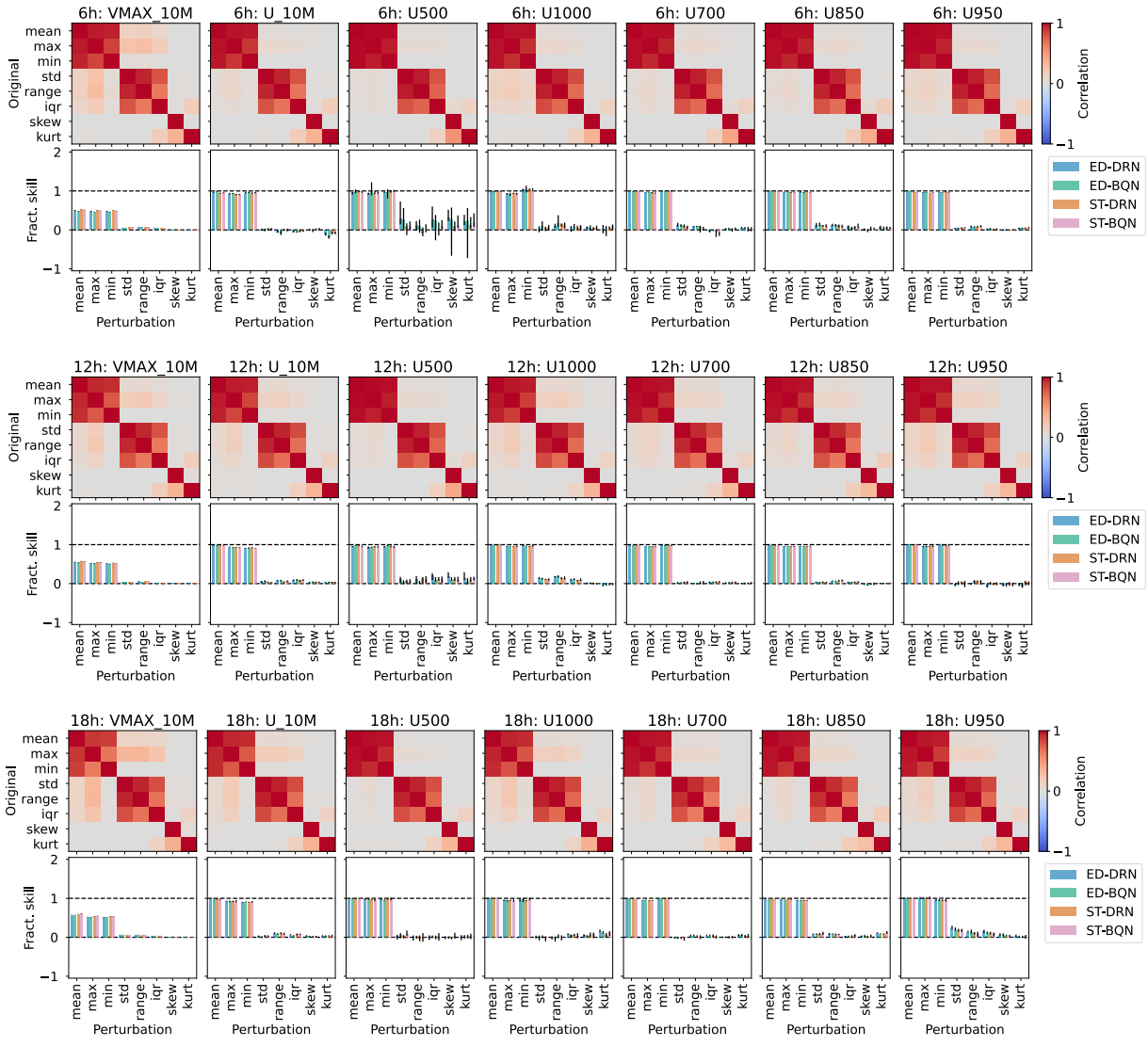

FIG. 8. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 1). Same as Fig. 4 in the main text.

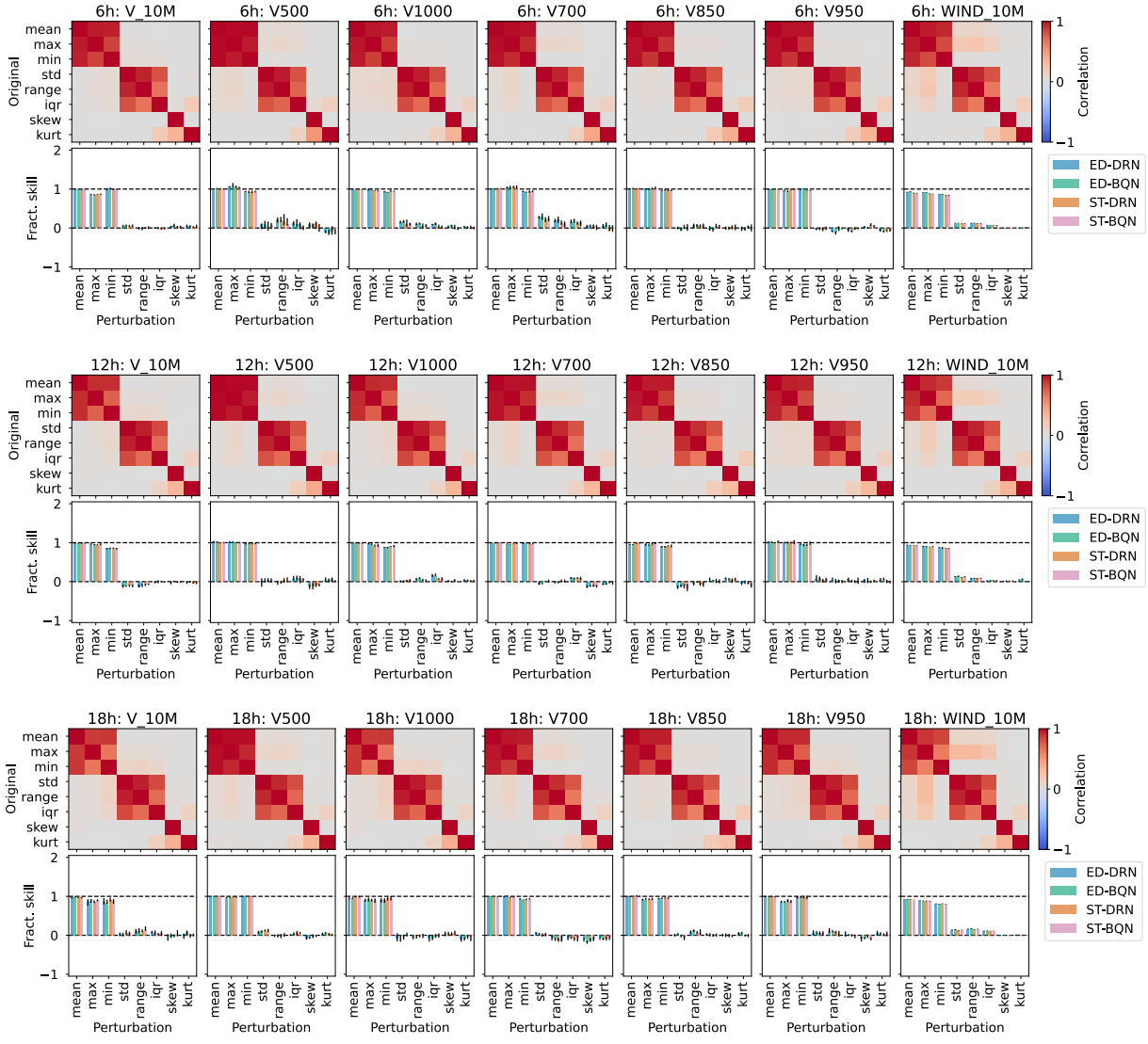

FIG. 9. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 2). Same as Fig. 4 in the main text.

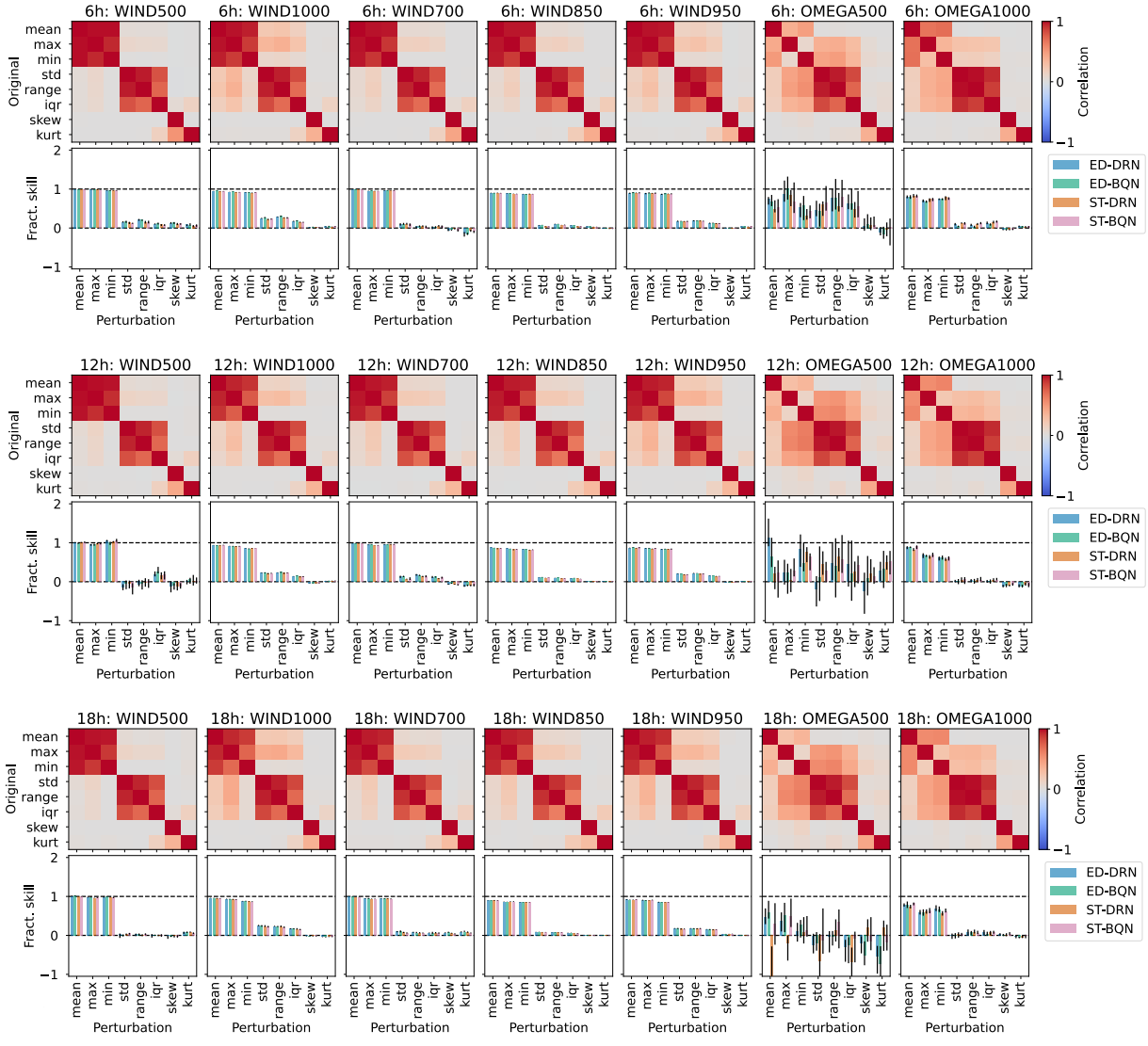

FIG. 10. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 3). Same as Fig. 4 in the main text.

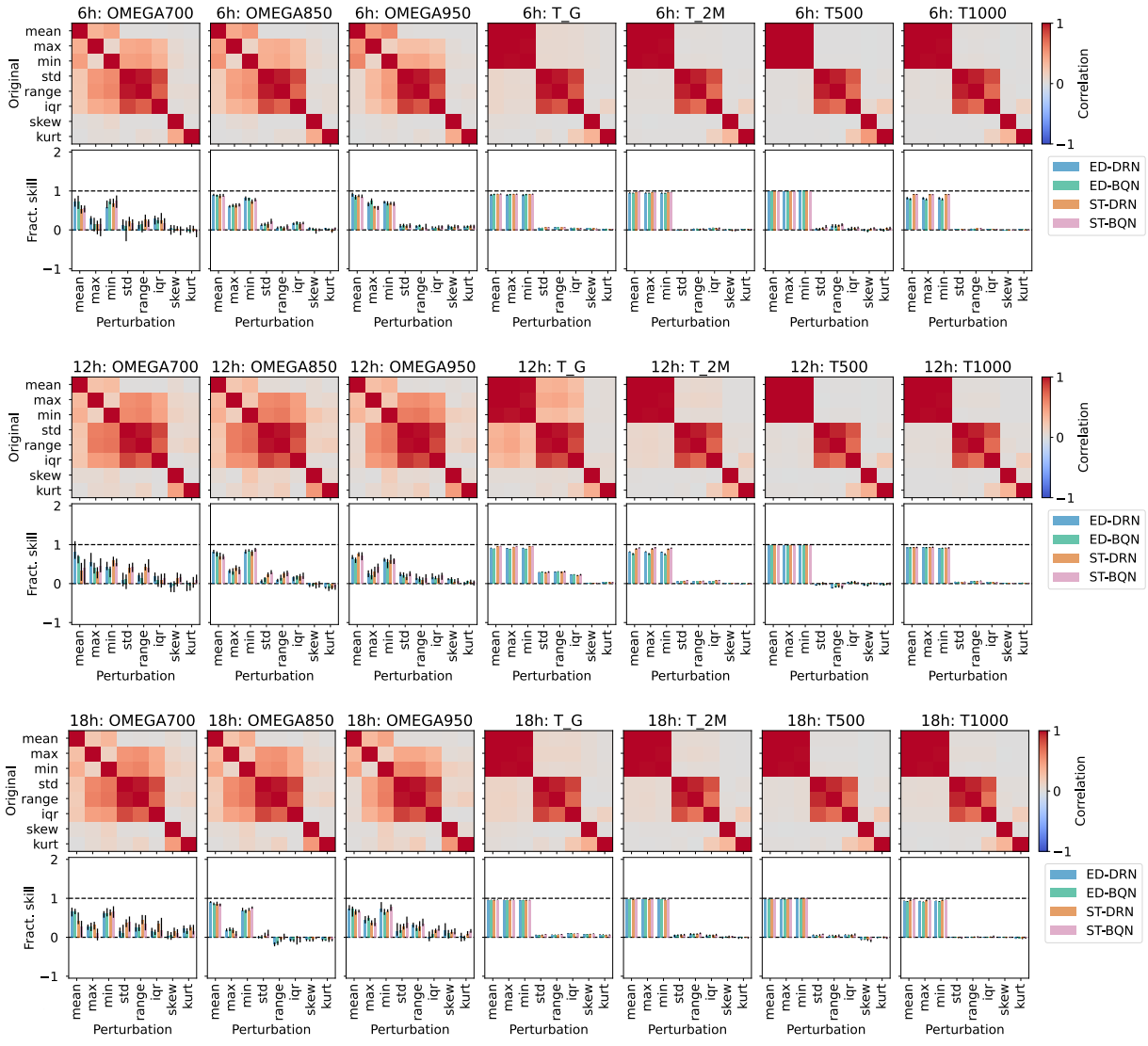

FIG. 11. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 4). Same as Fig. 4 in the main text.

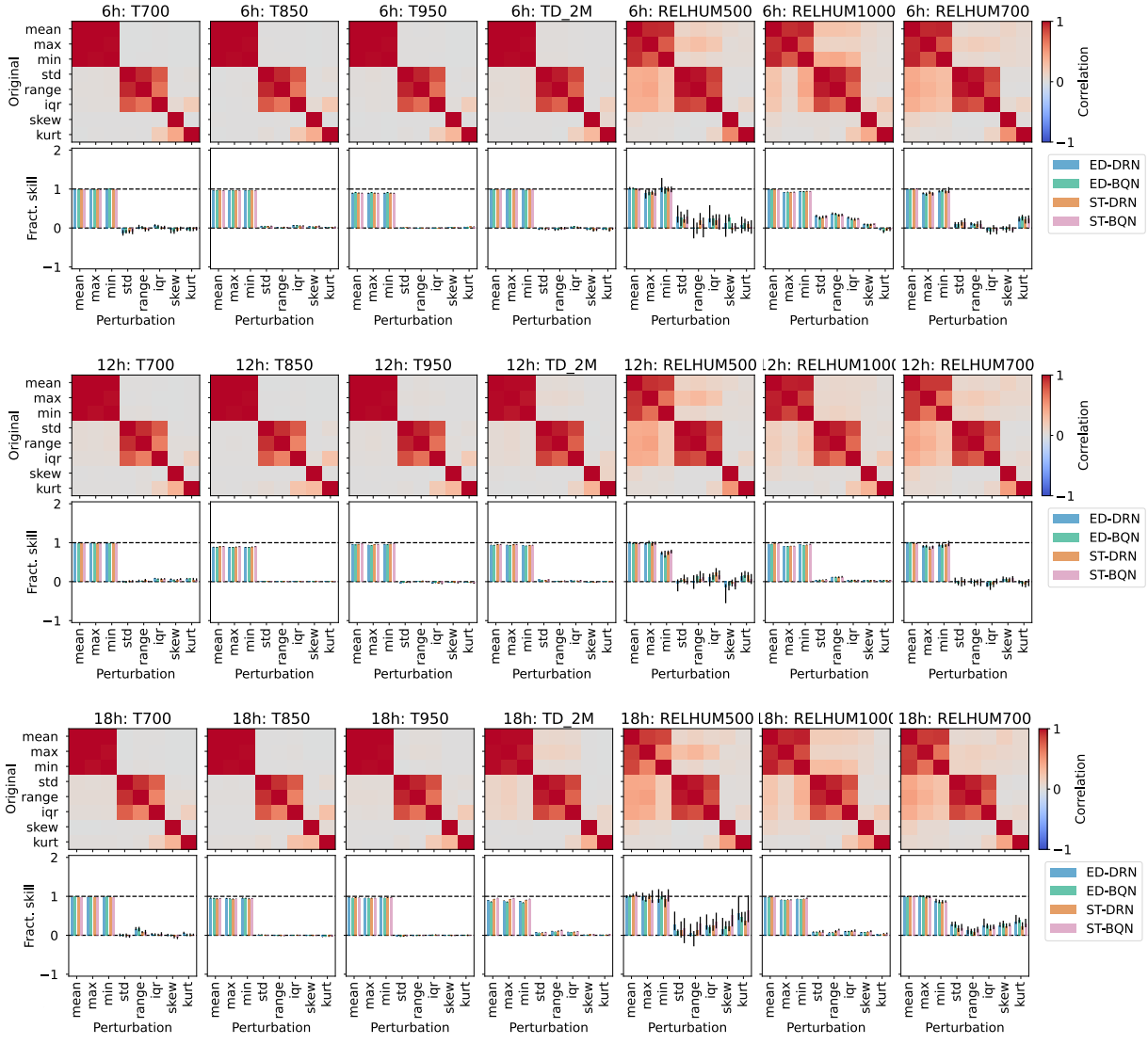

FIG. 12. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 5). Same as Fig. 4 in the main text.

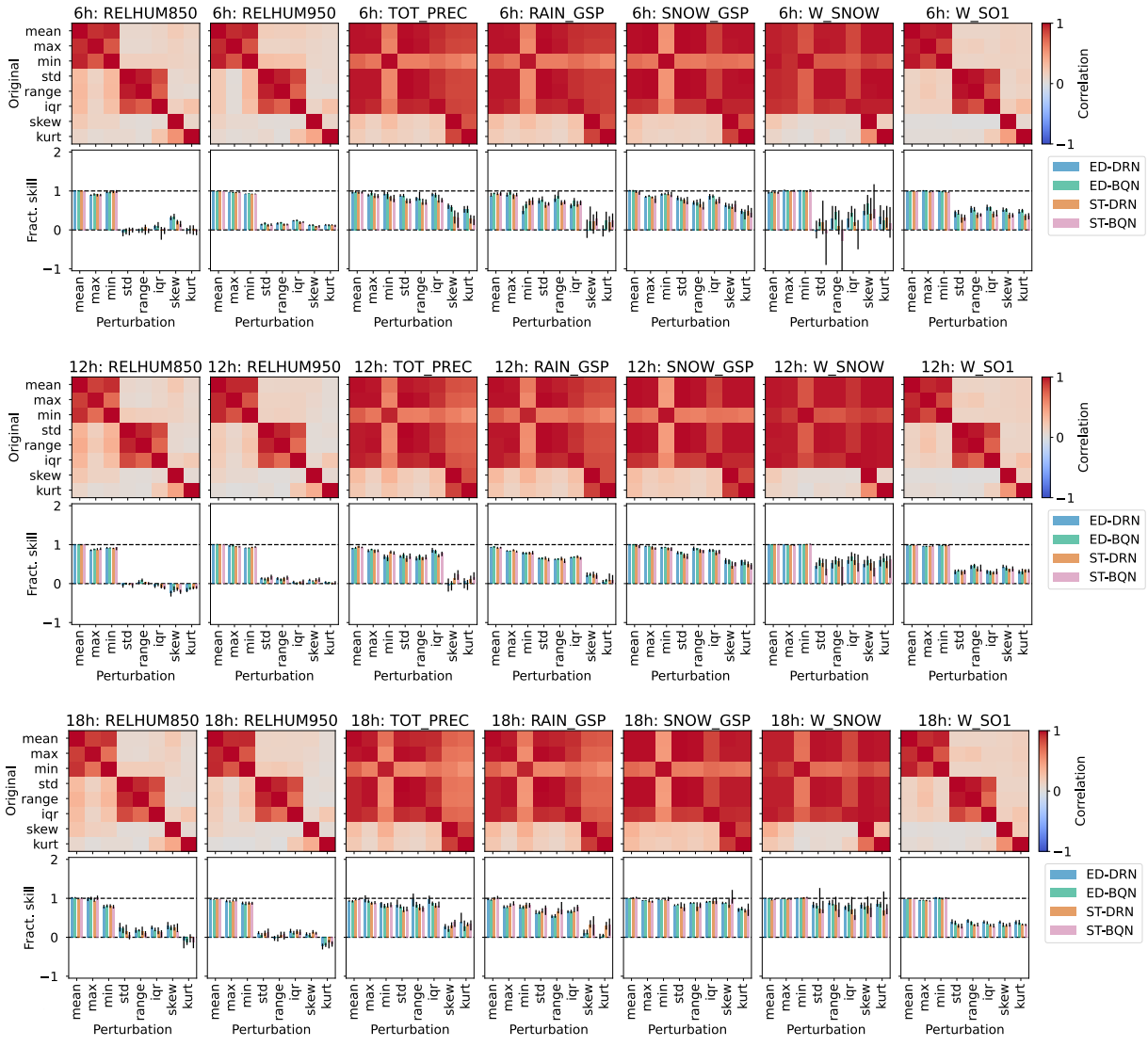

FIG. 13. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 6). Same as Fig. 4 in the main text.

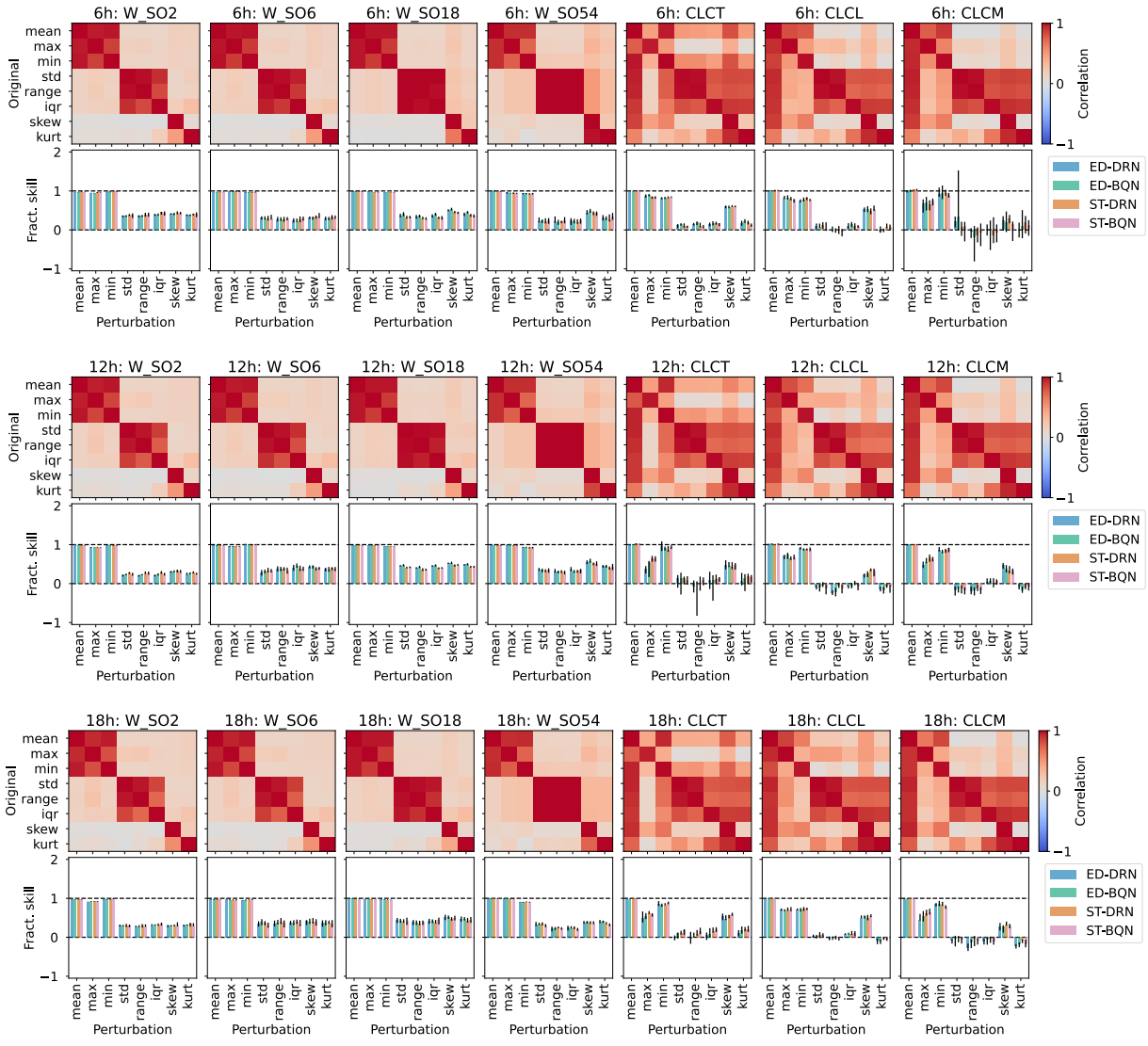

FIG. 14. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 7). Same as Fig. 4 in the main text.

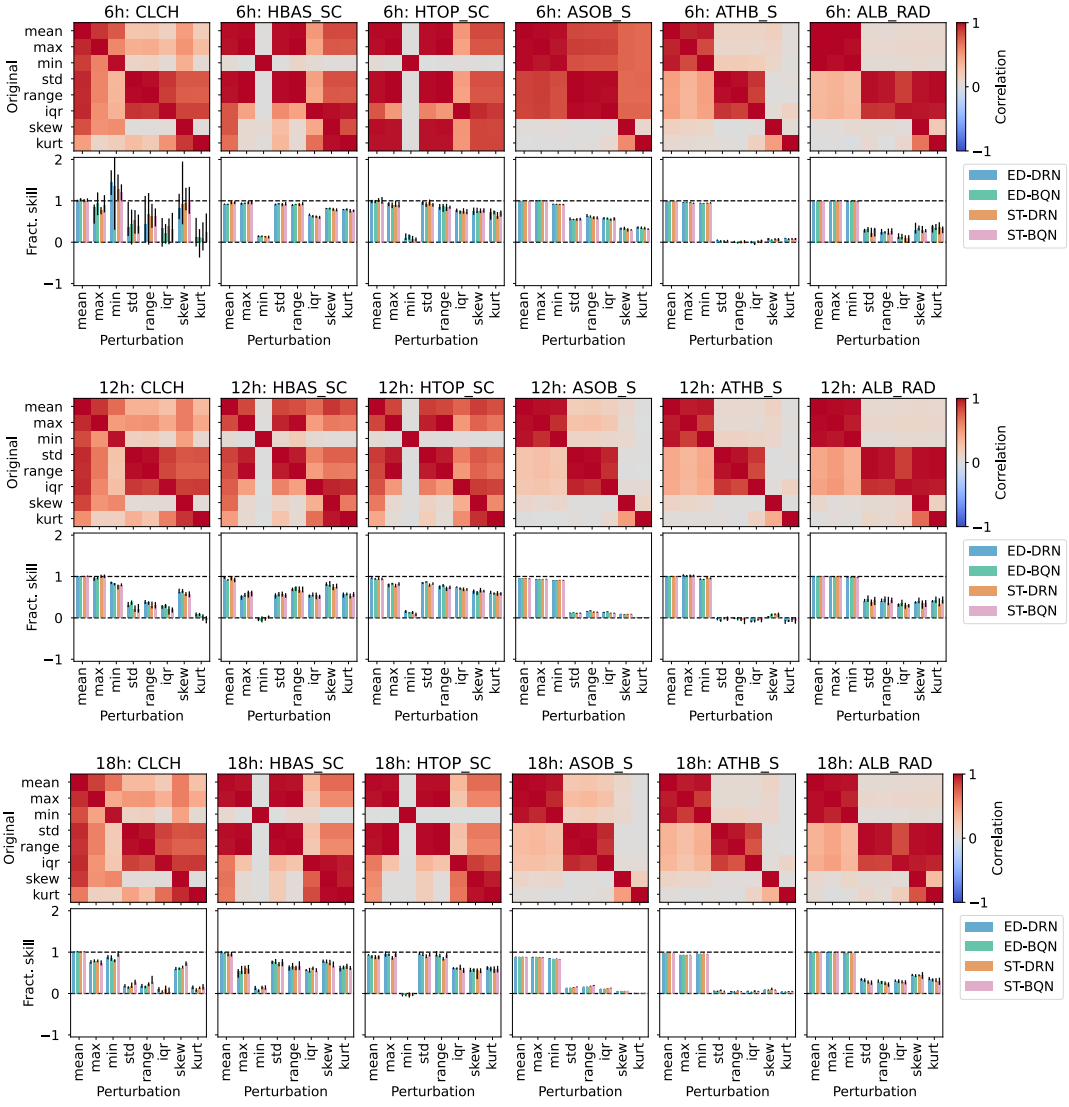

FIG. 15. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 8). Same as Fig. 4 in the main text.

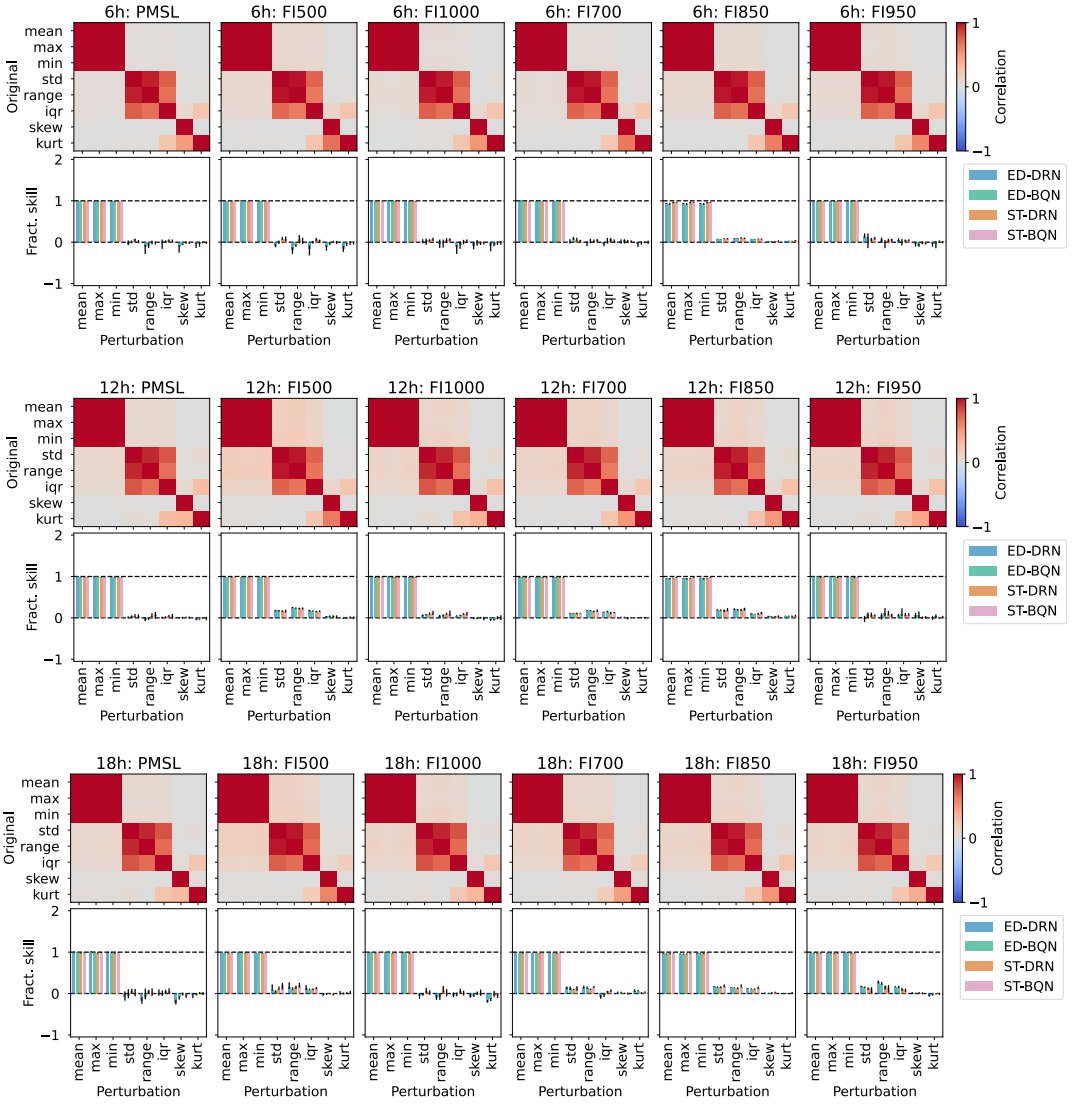

FIG. 16. Importance of ensemble-internal DOFs for wind-gust postprocessing (predictor batch 9). Same as Fig. 4 in the main text.

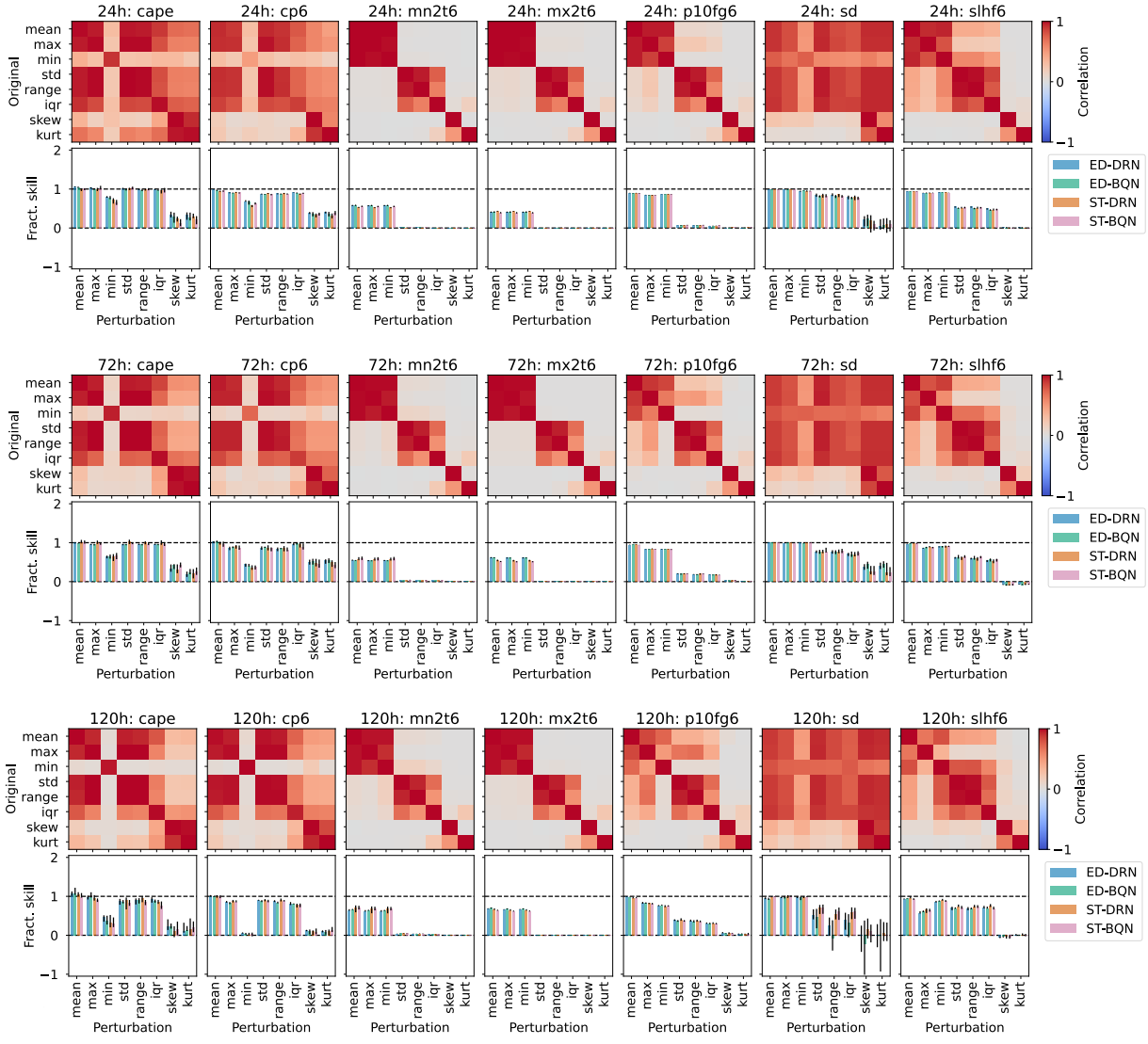

FIG. 17. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 1). Same as Fig. 5 in the main text.

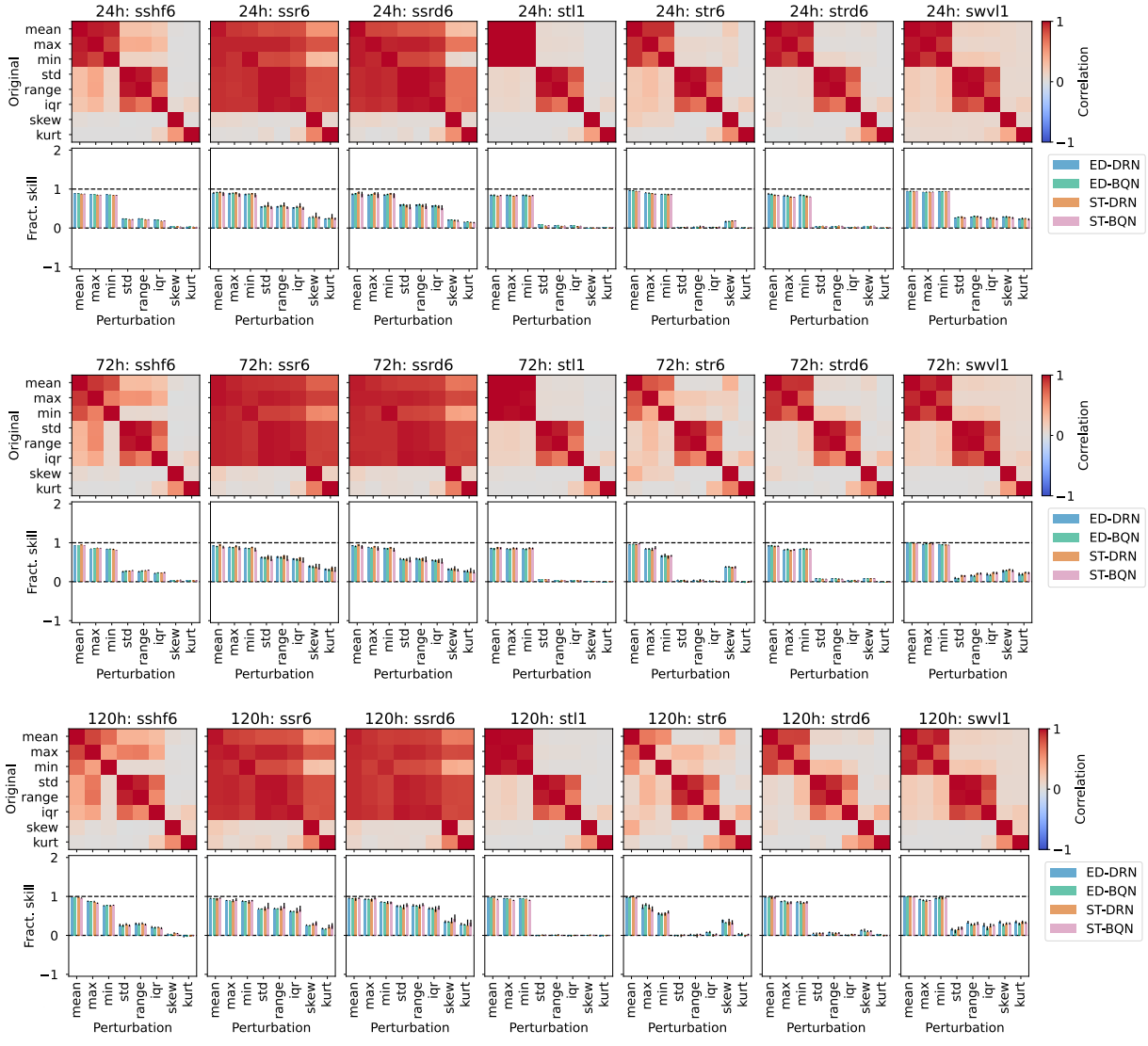

FIG. 18. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 2). Same as Fig. 5 in the main text.

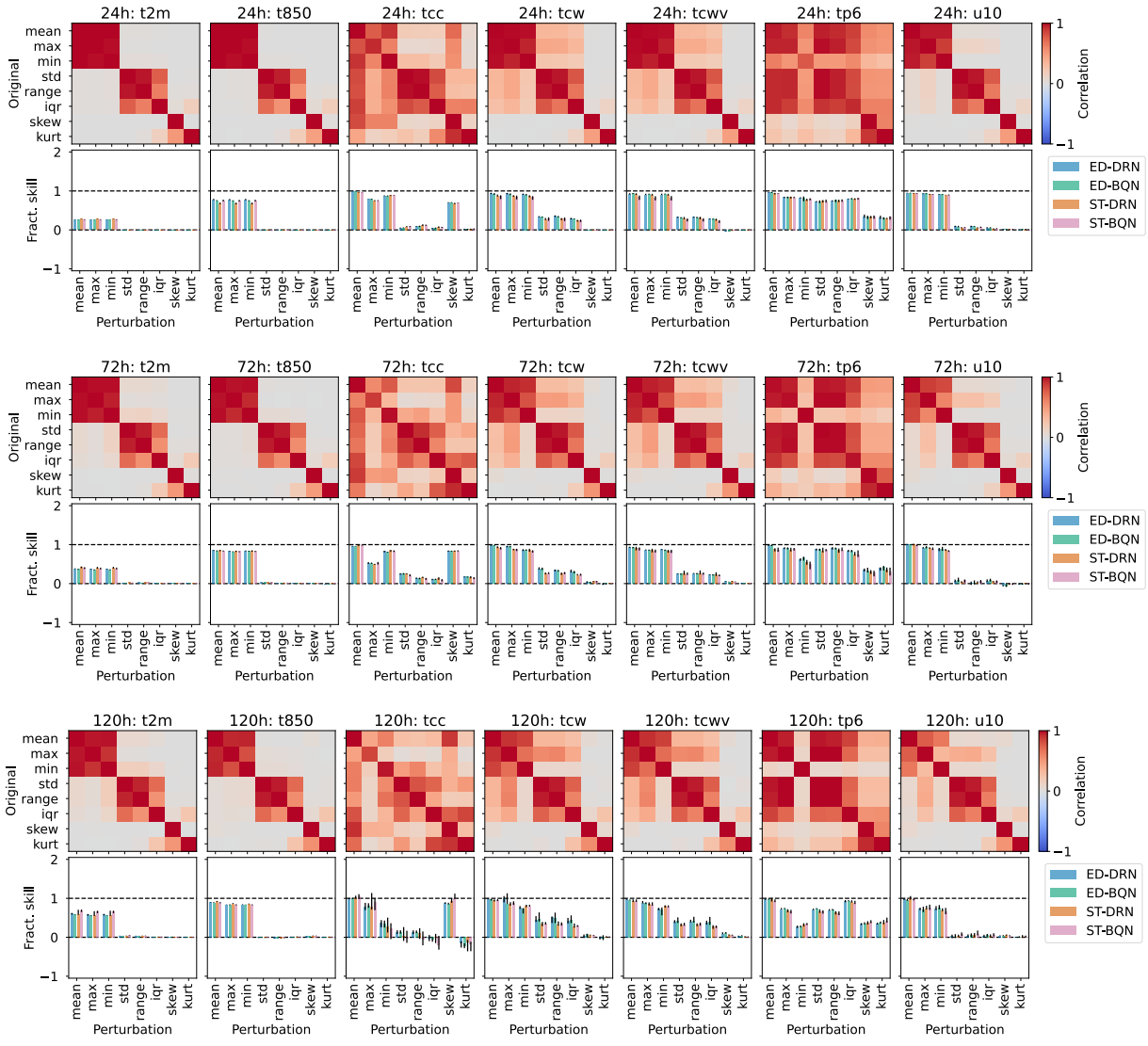

FIG. 19. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 3). Same as Fig. 5 in the main text.

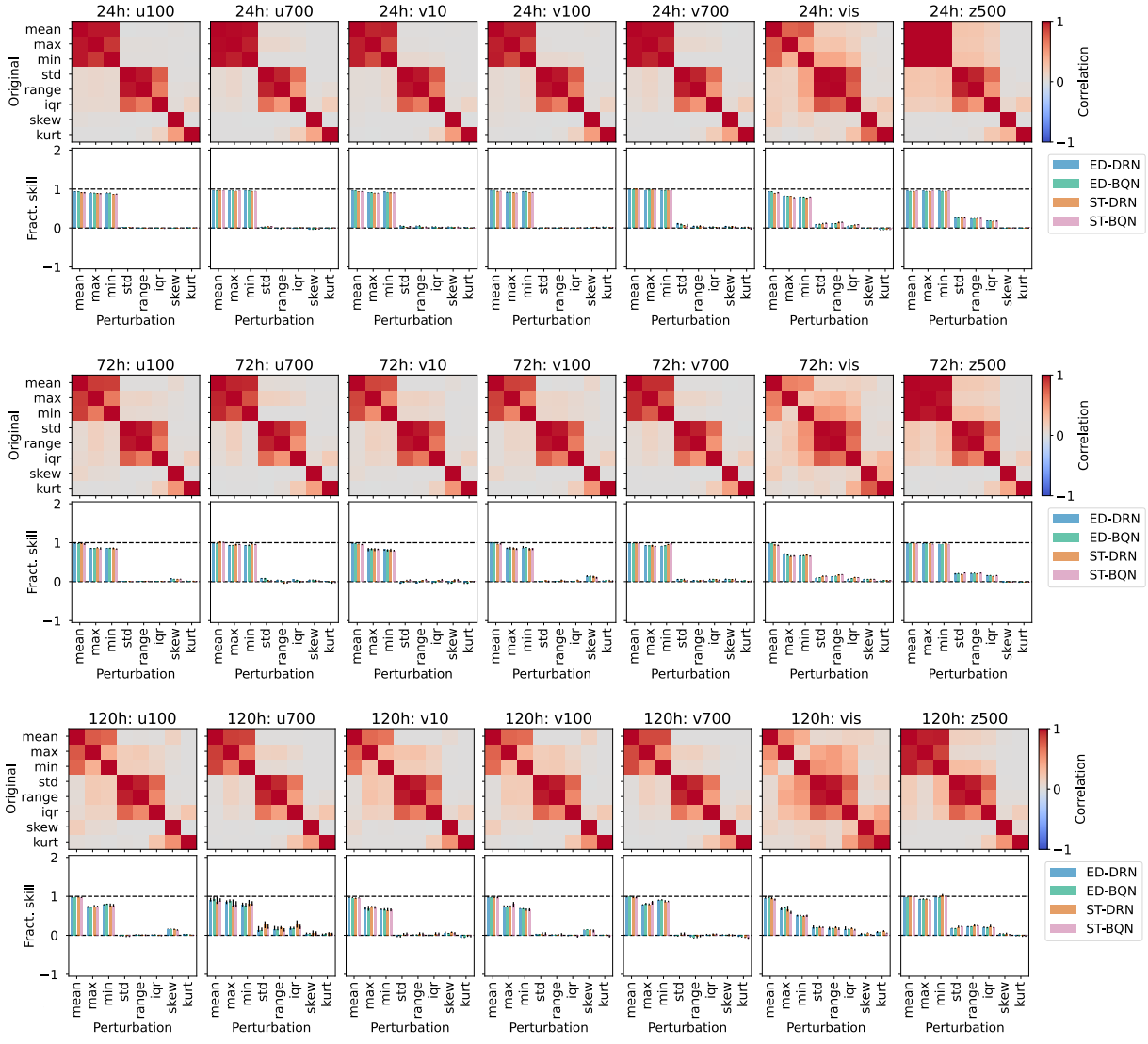

FIG. 20. Importance of ensemble-internal DOFs for temperature postprocessing (predictor batch 4). Same as Fig. 5 in the main text.