UNCERTAINTY-AWARE GENOMIC DEEP LEARNING WITH KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) have advanced predictive modeling for regulatory genomics, but challenges remain in ensuring the reliability of their predictions and understanding the key factors behind their decision making. Here, we introduce DEGU (Distilling Ensembles for Genomic Uncertainty-aware models), a method that integrates ensemble learning and knowledge distillation to improve the robustness and explainability of DNN predictions. DEGU distills the predictions of an ensemble of DNNs into a single model, capturing both the average of the ensemble's predictions and the variability across them, with the latter representing epistemic (or model-based) uncertainty. DEGU also includes an optional auxiliary task to estimate aleatoric, or data-based, uncertainty by modeling variability across experimental replicates. By applying DEGU across various functional genomic prediction tasks, we demonstrate that DEGU-trained models inherit the performance benefits of ensembles in a single model, with improved generalization to out-of-distribution sequences and more consistent explanations of cis-regulatory mechanisms through attribution analysis. Moreover, DEGU-trained models provide calibrated uncertainty estimates, with conformal prediction offering coverage guarantees under minimal assumptions. Overall, DEGU paves the way for robust and trustworthy applications of deep learning in genomics research.

1 INTRODUCTION

Deep neural networks (DNNs) have demonstrated strong performance in predicting the results of functional genomic experiments directly from DNA sequences (Avsec et al., 2021a; Chen et al., 2022; Dudnyk et al., 2024). By approximating experimental assays, these DNNs enable virtual experiments that explore the functional effects of genomic sequence perturbations. In these applications, high-performing DNNs serve as black-box *in silico* oracles or scoring functions, mapping DNA sequence inputs to a target molecular phenotype such as gene expression or chromatin accessibility. These models have the potential to improve hypothesis generation and guide more optimal experimental design, setting the stage for efficient AI-guided biological discovery.

However, these downstream applications assume that DNNs maintain their predictive performance even when the statistical properties of the input data differ from those seen during training, a phenomenon known as a covariate shift (Shimodaira, 2000). Model generalization, or the ability of models to make accurate predictions on these previously unseen out-of-distribution data points, is often assessed using held-out sequences from the same experiment that generated the training data. Although these held-out sequences come from different genomic regions, they are typically similar in genomic composition to the training data due to evolutionary constraints. Consequently, a model's performance on these in-distribution sequences may not accurately reflect their true generalizability. In fact, recent studies examining state-of-the-art genomic DNNs, such as Enformer (Avsec et al., 2021a), have shown that while these models generalize well to single nucleotide variant effects within cis-regulatory elements (Karollus et al., 2023; Seitz et al., 2024), they struggle to predict the effects of population-level genetic variations that involve only a few, sparse mutations (Sasse et al., 2023; Huang et al., 2023). This raises an important question: which predictions can we trust? Our inability to quantitatively assess the confidence of genomic DNN predictions undermines their reliability in downstream applications. Thus, addressing and quantifying uncertainty remains a key challenge for the field.

One approach to quantifying uncertainty involves training *deep ensembles*, or ensembles of DNNs wherein each model typically shares the same architecture but differs in their randomly initialized parameters. This variation means that the variability across their predictions can serve as an empirical measure of epistemic uncertainty, or model uncertainty due to limited data (Dietterich, 2000; Lakshminarayanan et al., 2017). For example, a recent study leveraged deep ensembles of genomic DNNs to investigate relationships between small sequence perturbations (e.g., expression quantitative-trait loci) and uncertainty in predictions (Bajwa et al., 2024). These *deep ensembles* (Lakshminarayanan et al., 2017) also improve predictive performance by averaging the predictions across the constituent models in the ensemble, with each model capturing a different aspect of the data. Averaging reduces individual model errors and balances biases, leading to more accurate and robust predictions than any single model alone. Ensembling has indeed proven to be an effective strategy to improve predictive performance for genomic DNNs (Malina et al., 2022; Agarwal et al., 2023; Linder et al., 2023b; He & Danko, 2024). Deep ensembles also provide more reliable post hoc explanations when averaging the attribution maps from each model in the ensemble. These attribution maps assign importance scores to nucleotides in a given sequence, revealing sequence motifs that are functionally relevant for the model's predictions (Gyawali et al., 2022; Majdandzic et al., 2023; Seitz et al., 2024; Novakovsky et al., 2023).

Despite these advantages, deep ensembles face several challenges that limit their practicality. One major issue is the increased computational overhead required to train and deploy multiple models, making large-scale inference tasks such as genome-wide variant effect predictions or extensive *in silico* experiments computationally expensive. The need to manage and maintain multiple models also adds substantial complexity to implementation, creating scalability challenges, especially as model architectures continue to trend towards greater size and complexity (Avsec et al., 2021a; Linder et al., 2023b; Zhou, 2022; Karbalayghareh et al., 2022; Hingerl et al., 2024; Lal et al., 2024). Furthermore, while deep ensembles capture epistemic uncertainty, they fail to account for *aleatoric uncertainty* (Hüllermeier & Waegeman, 2021; Der Kiureghian & Ditlevsen, 2009), the irreducible noise that stems from the technical and biological variability inherent in sequencing data.

To address these limitations, we introduce DEGU (Distilling Ensembles for Genomic Uncertainty-aware models), a method that combines ensemble learning and knowledge distillation (Hinton et al., 2015) to improve the robustness and explainability of DNN predictions. DEGU leverages ensemble distribution distillation (Malinin et al., 2019), a variant of knowledge distillation that focuses on learning the distribution of predictions from the ensemble rather than individual point estimates. This is accomplished by training a single student model in a multitask fashion to perform two primary tasks: 1) predict the mean of the ensemble's predictions, and 2) estimate the corresponding epistemic uncertainty based on the variability across the ensemble's predictions. DEGU can also incorporate an optional auxiliary prediction task for aleatoric uncertainty, estimated from the variability observed across experimental replicates.

By applying DEGU to different DNNs across various functional genomics prediction tasks, we found that distilled models exhibit improved generalization and enhanced robustness in their attribution maps compared to standard training methods. Furthermore, DEGU-distilled models accurately predict epistemic uncertainty. Together, DEGU provides the efficiency of a single model during inference while preserving the performance and robustness of deep ensembles, with the added benefit of generating calibrated uncertainty estimates.

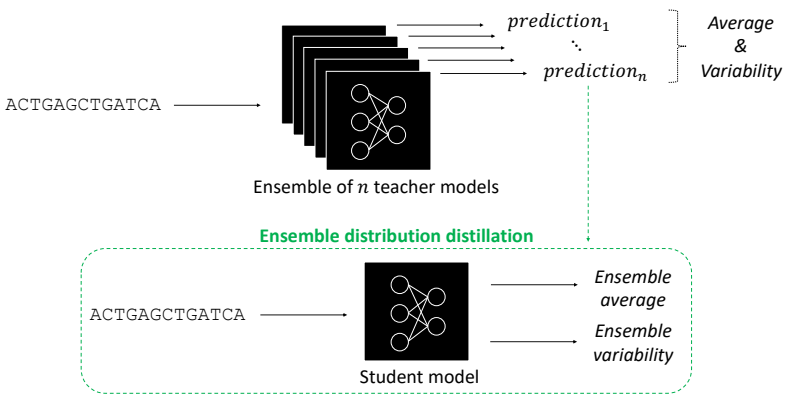


Figure 1: Schematic of ensemble distribution distillation with DEGU.

2 RESULTS

2.1 DEGU: DISTILLING THE KNOWLEDGE OF ENSEMBLES INTO A SINGLE MODEL

DEGU employs ensemble distribution distillation (Malinin et al., 2019) to transfer the collective knowledge from an ensemble of models, which we refer to as teacher models, to a single student model (Fig. 1). The process begins with the creation of a teacher ensemble composed of multiple DNNs trained independently with different random initializations. Through knowledge distillation, the student model learns the distribution of predictions from the teacher ensemble by performing multiple tasks concurrently: 1) predicting the mean of the ensemble's predictions and 2) predicting the variability across the ensemble's predictions. This assumes that the predictions across the teacher ensemble follow a normal distribution, where the mean is used to distill an ensemble's predictions and the standard deviation reflects epistemic uncertainty. When at least three experimental replicates are available, the student model can optionally be trained to predict the aleatoric uncertainty (Kendall & Gal, 2017) as well, which is approximated by the variability observed across replicates in the training data. Our multitask learning approach ensures that DEGU captures the distribution of the predictions of the ensemble along with variability inherent in the data. Altogether, DEGU retain the performance and robustness advantages of deep ensembles while significantly reducing computational overhead during downstream inference tasks.

2.2 DEGU APPROXIMATES THE PREDICTIVE PERFORMANCE OF DEEP ENSEMBLES

We applied DEGU to various genomic sequence DNNs for diverse datasets: fly enhancer activity (STARR-seq) for developmental (Dev) and housekeeping (Hk) promoters (de Almeida et al., 2022), human cis-regulatory sequence activity (lentiMPRA) for K562 and HepG2 cells (Agarwal et al., 2023), and base-resolution ATAC-seq profiles from a human cell line (Buenrostro et al., 2015). For each application, we constructed a teacher ensemble consisting of 10 models using established architectures suited to each task: multi-task DeepSTARR (de Almeida et al., 2022) for fly enhancers, single-task ResidualBind (Koo et al., 2021; Tang et al., 2024) for lentiMPRA (Agarwal et al., 2023), and a standard convolutional neural network (CNN) for base-resolution ATAC-seq profiles (Toneyan et al., 2022). Student DNNs with the same architecture as their respective teacher models were then trained with DEGU's ensemble distribution distillation procedure (see Appendix B).

Strikingly, the distilled student models outperformed the teacher models with standard training despite sharing the same architecture, particularly in low data regimes (Fig. 2a-b, Appendix A Figs. 1 & 2a). For example, distilling DeepSTARR with only 25% of the STARR-seq training data yielded performance comparable to standard training on the full dataset for both developmental and housekeeping promoters (Fig. 2a-b). The benefits of DEGU were more nuanced for lentiMPRA data, with substantial gains in K562 and more modest improvements in HepG2 (Appendix A Fig. 1). For base-resolution ATAC-seq profiles, the performance of the distilled DNNs was comparable to that of standard-trained DNNs (Appendix A Fig. 2a), albeit with a slight decrease in performance.

We observed additional performance gains when the performance of individual models within the teacher ensemble were improved (Appendix A Fig. 3). In this scenario, all teacher and student models were trained with evolution-inspired data augmentations generated with EvoAug (Lee et al., 2023; Yu et al., 2024). This yielded performance gains across all models (Appendix A Fig. 3). Moreover, as the number of models in the teacher ensemble increased, the ensemble's predictive performance improved and plateaued around $n = 10$ models. The performance of the distilled models also plateaued but at smaller ensemble sizes of around $n = 5$ models (Appendix A Fig. 4). Notably, the performance gap between the ensemble and the distilled models widened as the ensemble size increased beyond this range.

2.3 DEGU IMPROVES ATTRIBUTION ANALYSIS

Attribution methods assign an importance score to each nucleotide in a sequence, indicating how much that nucleotide contributes to the model's prediction or how sensitive the model's output is to changes at that nucleotide. Visualizing attribution scores as a sequence logo can reveal biologically meaningful patterns, such as transcription factor binding motifs (Avsec et al., 2021b; de Almeida et al., 2022; Koo & Ploenzke, 2021). However, attribution methods can be sensitive to local variations in the model's learned function, which can arise when fitting to noise in the data.

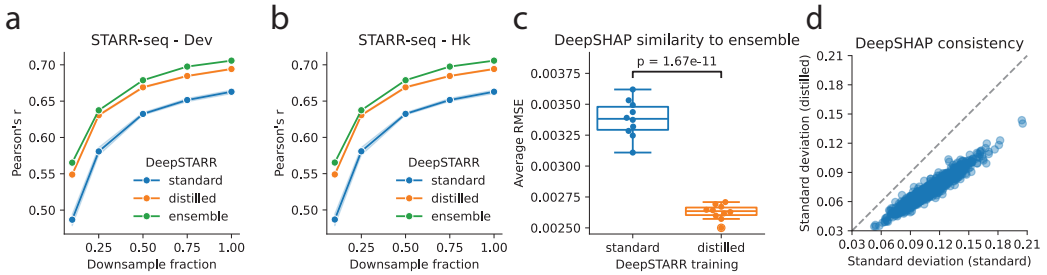


Figure 2: Evaluating advantages of DEGU on DeepSTARR. (**a**,**b**) Performance of models trained on random subsets of STARR-seq data for (**a**) developmental (Dev) and (**b**) housekeeping (Hk) promoters. (**c**) Average RMSE between DeepSHAP scores for models with standard training (blue) and DEGU distillation (orange) compared to the average scores across the teacher ensemble. RMSE was calculated for $n = 1000$ high-activity test sequences. P-values indicate independent two-sided t-tests (average RMSE) and paired two-sided t-tests (standard deviation). Box plots represent $n = 10$ models trained with different random initializations. (**d**) Scatterplots comparing standard deviation of DeepSHAP scores across models trained with DEGU distillation and standard training ($n = 10$).

This variability may not affect a model's ability to generalize to unseen data (i.e., benign overfitting (Bartlett et al., 2020)), but it can lead to inconsistent explanations (Han et al., 2022; Wang et al., 2020; Alvarez-Melis & Jaakkola, 2018), making it difficult to distinguish biologically relevant patterns from spurious importance scores caused by non-biological fluctuations (Seitz et al., 2024; Majdandzic et al., 2022). By averaging attribution maps across an ensemble of models, some of these fluctuations may be reduced, leading to more robust explanations (Majdandzic et al., 2023; Gyawali et al., 2022). We hypothesized that DEGU-distilled student models, which better approximate the ensemble function, would produce more interpretable and robust attribution maps with stronger motif signals compared to models with standard training.

To test this hypothesis, we generated attribution maps using DeepSHAP (Lundberg & Lee, 2017) and Saliency Maps (Simonyan et al., 2013) for models trained with DEGU and with standard training. A visual comparison revealed that the attribution maps from the DEGU-distilled student models displayed more identifiable transcription factor motifs, such as GATA and AP-1, compared to models with standard training (Appendix A Fig. 5).

Assuming that ensemble-averaged attribution maps best reflect underlying biology, we compared the attribution maps generated by averaging across the ensemble and those from individual models trained with either standard training or DEGU distillation (see Appendix B). We found that across all prediction tasks evaluated, the attribution maps produced by distilled models were more closely aligned with the ensemble-averaged attribution maps than those generated by models with standard training (Fig. 2c, Appendix A Fig. 6, Appendix A Fig. 2b).

Additionally, attribution maps generated by different distilled student models were significantly more consistent compared to models with standard training (Fig. 2c) across different models, datasets, and attribution methods (Appendix A Fig. 7, Appendix A Fig. 2b). However, variability across attribution maps from different models can stem from variability in the magnitude of the attribution scores and/or the sequence content (i.e., distinct cis-regulatory mechanisms). To control for variability in attribution score magnitude, we normalized the attribution scores for each sequence. We found that distilled models remained more consistent than the models with standard training (Appendix A Fig. 7), suggesting that the distilled models offer more robust mechanistic insights through their attribution maps.

2.4 DEGU IMPROVES GENERALIZATION UNDER COVARIATE SHIFTS

Most downstream applications of genomic DNNs require them to generalize well under covariate shifts, especially when making predictions for sequence perturbations that were not represented in the training data. Ensembles are typically expected to improve out-of-distribution (OOD) generalization (Arpit et al., 2022; Lakshminarayanan et al., 2017) because they aggregate variable predictions for a given input, thereby smoothing out arbitrary behavior in regions with limited or no data

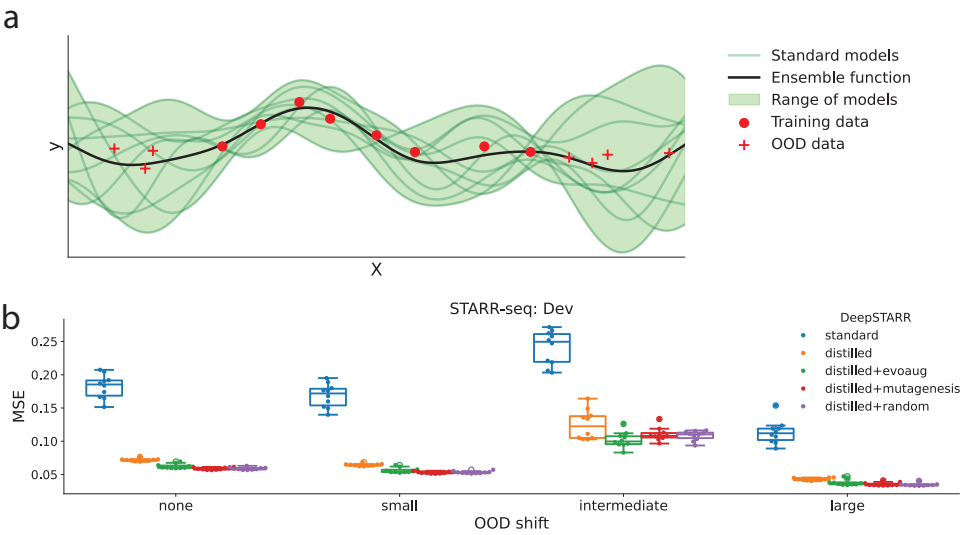


Figure 3: DeepSTARR OOD generalization performance. (**a**) Schematic illustrating an ensemble of functions learned from different random initializations (green lines) on training data alongside the averaged ensemble function (black line). Generating labels for OOD data points using the ensemble's predictions stabilizes the distilled model's function approximation in OOD regions. (**b**) MSE between the teacher ensemble (trained with EvoAug) and individual models for Dev promoter activity across different training procedures: standard training (blue), DEGU-distillation (orange), and DEGU-distillation with dynamic EvoAug mutagenesis (green), partial random mutagenesis (red), and randomly shuffled sequences (purple), evaluated for for sequences with varying degrees of distribution shift. Boxplots represent $n = 10$ models trained with different random initializations.

(Fig. 3a). Thus, we hypothesized that DEGU-distilled models, which approximate the ensemble's function, would also generalize better to OOD sequences. However, systematically assessing OOD generalization is challenging due to the limited availability of appropriate OOD data; that is, experimental measurements in the same biological system for sequences with matched levels of genetic variability as the downstream task. Instead, we used a proxy for OOD generalization by evaluating how closely the distilled models approximated the teacher ensemble's behavior under varying levels of simulated covariate shift. Specifically, we created three new variants of test sequences from the original STARR-seq test sequences, each simulating a different degree of distribution shift: (1) partial random mutagenesis at a rate of 0.05 at each position of the sequence to introduce a small shift, (2) evolution-inspired mutagenesis provided by EvoAug (Lee et al., 2023; Yu et al., 2024) for an intermediate shift, and (3) randomly shuffled sequences for a large shift (see Appendix B). The small shift introduced by partial random mutagenesis likely preserved most key motifs and overall sequence function. In contrast, the intermediate shift generated through evolution-inspired mutagenesis created more substantial compositional rearrangements in regulatory sequences. The large shift, created by randomly shuffling the sequences, likely disrupted and inactivated many functional regions, resulting in lower predicted regulatory activity overall (Appendix A Fig. 8).

We then calculated the mean-squared error (MSE) between each model's predictions and an ensemble average of $n = 10$ EvoAug-trained DeepSTARR models, which we treat as an in silico oracle. As expected, the distilled DeepSTARR models provided consistently closer approximations of the teacher ensemble over standard-trained DeepSTARR models (Fig. 3b, Appendix A Fig. 9). Interestingly, we observed lower MSE for inference on randomly shuffled sequences, possibly due to the lower overall activity levels following this large covariate shift (Appendix A Fig. 8).

Next, we explored whether training with OOD data could improve model generalization (Fort et al., 2021; Wilson & Izmailov, 2020; Hoffmann et al., 2021). Specifically, we hypothesized that introducing OOD sequences with ensemble-generated labels would help the model better approximate the ensemble function in regions where training data is sparse. To test this, we applied the same transformations used to simulate covariate shifts to each minibatch of sequences during training and used the teacher ensemble to generate corresponding labels for these transformed sequences (see Appendix B. The augmented sequences and their new target values replaced the original training data, allowing us to train distilled DeepSTARR models with more diverse data points.

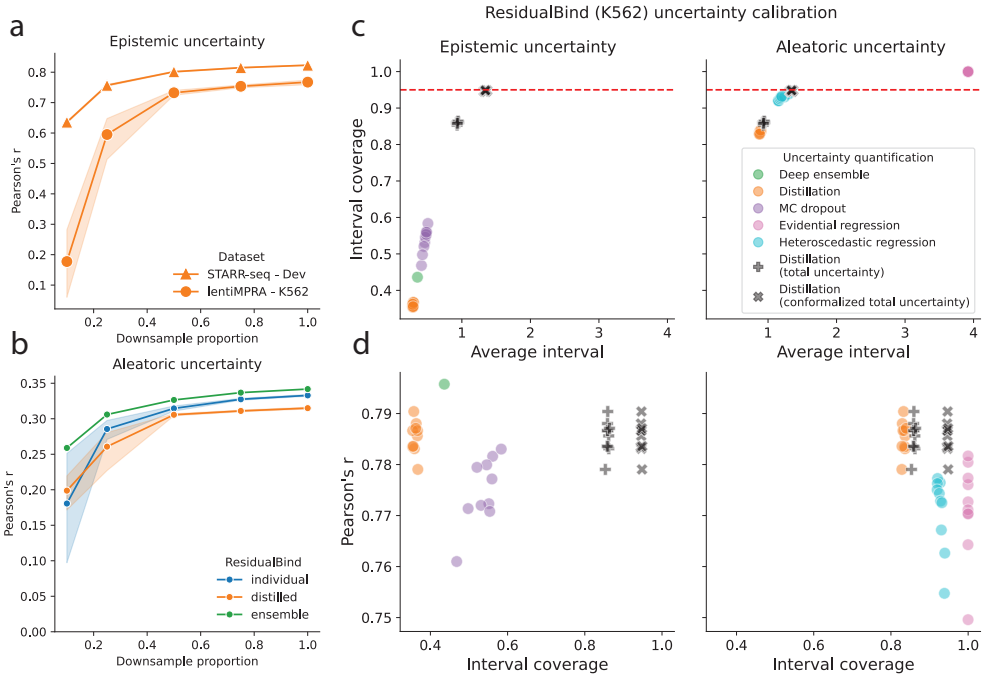


Figure 4: Evaluating DEGU uncertainty estimation. (**a**,**b**) Predictive performance for (**a**) epistemic and (**b**) aleatoric uncertainty output heads of models trained with standard training (blue), DEGU-distillation (orange), and the teacher ensemble (green). Markers represent average Pearson's r across $n = 10$ models and shaded region indicates 95% confidence interval. Results shown for the Dev epistemic uncertainty output head of DEGU-distilled DeepSTARR and the uncertainty output heads of ResidualBind models trained on K562 lentiMPRA data. (**c**,**d**) Scatterplots of (**c**) prediction interval coverage probability and (**d**) predictive accuracy versus average interval size for different epistemic (left) and aleatoric (right) uncertainty quantification methods for ResidualBind models trained on K562 lentiMPRA data. Red dashed line indicates calibration with a 95% interval coverage probability. Each uncertainty quantification method is represented by $n = 10$ dots, each indicating a model with different initializations, with the exception of deep ensemble ($n = 1$).

Training distilled student models with dynamic data augmentations provided consistent, though modest, performance gains on the original test set (Appendix A Fig. 10). Furthermore, these augmentations improved the student models' function approximation to the teacher ensemble under different levels of covariate shift, with slightly better performance when the augmentations closely matched the degree of target covariate shift (Fig. 3b, Appendix A Fig. 9). These findings highlight the importance of incorporating OOD training sequences to improve model reliability in downstream applications that require robust generalization to covariate shifts.

2.5 DEGU PROVIDES CALIBRATED ESTIMATES OF TOTAL UNCERTAINTY

A key advantage of deep ensembles is their ability to quantify epistemic uncertainty from the variability of predictions across the ensemble. To evaluate DEGU's ability to quantify this type of uncertainty, we compared the standard deviation of predictions from the teacher ensemble with the predicted standard deviations from the DEGU-distilled student models. Surprisingly, the epistemic uncertainty estimates from the distilled student models were strongly correlated with the variation observed across the teacher ensemble's predictions, suggesting that DEGU distillation effectively captures epistemic uncertainty (Fig. 4a, Appendix A Figs. 11a, 2a, & 12). We repeated this analysis using log-variance as the measure of variation and observed similar results (Appendix A Fig. 13).

However, relying solely on epistemic uncertainty can lead to overconfident predictions (Bajwa et al., 2024) as it does not capture the full spectrum of predictive uncertainty. DEGU addresses this limitation by training models to also predict aleatoric uncertainty, which is estimated from the variability across experimental replicates (when sufficient replicates are available; see Appendix B). We demonstrated this approach using the lentiMPRA (Agarwal et al., 2023) and ATAC-seq profile

datasets, both of which had three experimental replicates. We found that predicting aleatoric uncertainty from sequence data remains challenging (Fig. 4b, Appendix A Figs. 11b & 2a), which is expected given that aleatoric uncertainty represents irreducible noise.

While accurately predicting aleatoric uncertainty is difficult, ensuring that uncertainty estimates are well-calibrated is more important. To evaluate this, we analyzed prediction interval coverage probability. This metric measures how often the true target value for a given sequence falls within the confidence interval estimated for its corresponding predictions of activity and uncertainty. A well-calibrated model should achieve a coverage probability equal to the confidence interval percentage (95% in this analysis) while minimizing the size of the interval. We benchmarked the calibration of uncertainty estimates from DEGU-distilled models against various other uncertainty quantification strategies (see Appendix B, including those that capture epistemic uncertainty, such as Monte Carlo Dropout (MCDropout) (Gal & Ghahramani, 2016) and deep ensembles, as well as methods for estimating aleatoric uncertainty, such as deep evidential regression (Amini et al., 2020) and heteroscedastic regression (Nix & Weigend, 1994; Venkatesh & Thiagarajan, 2019). Methods accounting for aleatoric uncertainty generally achieved better calibration with smaller average interval sizes while methods relying solely on epistemic uncertainty exhibited severe under-calibration (Fig. 4c, Appendix A Fig. 14a). We also observed a trade-off between predictive accuracy and uncertainty calibration across different methods (Fig. 4d, Appendix A Fig. 14b).

Models trained with heteroscedastic regression were well-calibrated but suffered from lower predictive accuracy compared to DEGU-distilled models (Fig. 4c-d, Appendix A Fig. 14). Based on these observations, we investigated whether applying DEGU distillation to models trained with a heteroscedastic loss could reconcile this performance gap. However, this approach did not fully bridge the gap between heteroscedastic regression and DEGU (Appendix A Fig. 14).

To improve calibration, we applied conformal prediction (Barber et al., 2020; Papadopoulos et al., 2002; Vovk et al., 2005) to adjust the uncertainty estimates. This resulted in nearly perfect calibration for all uncertainty quantification methods evaluated (Fig. 4c, Appendix A Fig. 14), giving the distilled models the best overall performance balancing high predictive accuracy and well-calibrated uncertainty estimates.

2.6 Uncertainty-aware zero-shot variant effect generalization

Uncertainty quantification offers a valuable tool for informed decision making, particularly in scenarios where the reliability of model predictions is unclear, such as single-nucleotide variant effect prediction. Previously, we demonstrated that data augmentations improved function approximation to the ensemble (Appendix A Fig. 9), suggesting the potential for better generalization. To directly test this, we trained distilled ResidualBind models with dynamic augmentations on lentiMPRA data and evaluated their ability to predict single-nucleotide variant effects in matched cell types by comparing against experimental measurements from massively parallel reporter assays (MPRAs) (Critical Assessment of Genome Interpretation Consortium, 2024; Shigaki et al., 2019).

As expected, predicted variant effects from the distilled models generally showed stronger correlation with experimentally measured variant effects compared to models with standard training, with additional performance gains when distilled models were trained with data augmentations (Appendix A Figs. 15a & 16). Visual analysis indicated that aleatoric uncertainty predictions were greater in magnitude and more consistent across nucleotides at a given position, whereas epistemic uncertainty predictions exhibited greater variability across both nucleotides and positions (Appendix A Fig. 15b). Further analysis revealed that aleatoric uncertainty was higher for variant effects close to zero, while epistemic uncertainty increased as activity levels moved further from zero (Appendix A Fig. 19).

To investigate the relationship between uncertainty and predictive accuracy, we stratified model performance on total uncertainty, classifying predictions below this threshold as confident (see Appendix B. We focused on distilled ResidualBind models trained with random mutagenesis augmentations as they yielded the best overall performance (Appendix A Figs. 15a & 16). This uncertainty-based stratification revealed that the most confident predictions for variants in the *PKLR* locus from ResidualBind models trained on K562 were associated with higher predictive accuracy (Appendix A Fig. 15c), although these trends varied across different loci and models (Appendix A Fig. 18). These findings highlight the potential of uncertainty quantification to enhance the reliability and in-

terpretability of variant effect predictions, enabling more nuanced analysis and decision making in genomics research.

3 DISCUSSION

DEGU presents a simple and effective approach for harnessing the benefits of deep ensembles with just a single model, providing robust predictions alongside reliable uncertainty estimates. We demonstrate that DEGU-distilled models generally outperform models with standard training and provide more reliable post hoc explanations of cis-regulatory mechanisms through attribution analysis. This makes DEGU particularly well-suited for large-scale inference tasks such as genome-wide variant effect prediction and generating attribution maps for a large number of regulatory sequences.

A major strength of DEGU lies in its dual uncertainty estimation, addressing a key limitation of current genomic deep learning models. By simultaneously estimating both epistemic and aleatoric uncertainty, DEGU offers a comprehensive assessment of prediction reliability - an important factor in enhancing confidence in model predictions.

DEGU excels in estimating epistemic uncertainty, which was straightforward to train using prediction variability across the teacher ensemble. Estimating aleatoric uncertainty posed greater challenges due to inherent randomness in noisy data. To approximate this, we created a prediction task to learn noise across experimental replicates, yielding aleatoric uncertainty estimates with well-calibrated prediction intervals surpassing those based on epistemic uncertainty. We also evaluated heteroscedastic regression as an alternative approach for aleatoric uncertainty estimation as it avoids extra data processing steps and the inclusion of an additional prediction task. However, heteroscedastic regression showed lower predictive performance on functional activities and proved challenging to optimize due to an unstable loss function. Overall, aleatoric uncertainty estimates based on replicate variability combined with epistemic uncertainty estimates achieved the best balance between calibration and predictive accuracy. Nevertheless, heteroscedastic regression loss can serve as an alternative approach when the data contains insufficient replicates.

DEGU's ability to approximate the teacher ensemble's function suggests that distilled models generalize better under covariate shifts. Moreover, training with dynamic data augmentations further improved approximation of the ensemble. This proved effective for improving zero-shot predictions of single-nucleotide variant effects. Since most downstream applications of genomic DNNs involve varying degrees of covariate shifts, DEGU-distilled models are well suited to provide robust predictions with uncertainty estimates that reflect its confidence.

Our study focused solely on ensembles of models with the same architecture. It would also be worth applying DEGU to more compact student architectures to distill large-scale DNNs such as Enformer (Avsec et al., 2021a) and Borzoi (Linder et al., 2023a), both of which currently require high computational costs. Making these models more computationally efficient would reduce the need for extensive GPU resources, thereby democratizing access to state-of-the-art genomic prediction tools. Further improvements could be achieved by increasing the diversity of models within the ensemble and experimenting with different weighting methods for ensemble members.

In our study, DEGU primarily focused on approximating the ensemble's function, leaving opportunities for further improvement by incorporating a mixed knowledge distillation loss function that balances the use of real training data with the divergence between the ensemble and student models (Hinton et al., 2015). A related approach to DEGU is self-distillation (Zhang et al., 2022), where distillation occurs sequentially in an online manner. While self-distillation can offer benefits similar to ensemble distillation (Allen-Zhu & Li, 2020), it struggles to capture epistemic uncertainty as it relies on a single model's training path rather than the multiple function approximations provided by an ensemble. In the future, we plan to investigate the sequence features that contribute to uncertainty in genomic predictions through comprehensive attribution analyses on the uncertainty head.

DEGU's ability to provide insights into the confidence of its predictions represents an important step toward making deep learning models more reliable and trustworthy. While our demonstration of DEGU focused on deep learning models for regulatory genomics, this framework can be extended to other models in other domains of biology. As the field continues to evolve, uncertainty-aware models like DEGU will become essential for guiding research decisions and clinical applications, highlighting the importance of further refining and expanding these techniques.

REFERENCES

Vikram Agarwal, Fumitaka Inoue, Max Schubach, Beth K. Martin, Pyaree Mohan Dash, Zicong Zhang, Ajuni Sohota, William Stafford Noble, Galip Gürkan Yardimci, Martin Kircher, Jay Shendure, and Nadav Ahituv. Massively parallel characterization of transcriptional regulatory elements in three diverse human cell types, March 2023. URL https://www.biorxiv.org/content/10.1101/2023.03.05.531189v1. Pages: 2023.03.05.531189 Section: New Results.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods, 2018. URL https://arxiv.org/abs/1806.08049.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep Evidential Regression, November 2020. URL http://arxiv.org/abs/1910.02600. arXiv:1910.02600 [cs, stat].

Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 8265–8277. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/372cb7805eaccb2b7eed641271a30eec-Paper-Conference.pdf.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18 (10):1196–1203, 2021a.

Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat Genet*, 53(3): 354–366, March 2021b. ISSN 1546-1718. doi: 10.1038/s41588-021-00782-6. URL https://www.nature.com/articles/s41588-021-00782-6. Publisher: Nature Publishing Group.

Ayesha Bajwa, Ruchir Rastogi, Pooja Kathail, Richard W Shuai, and Nilah Ioannidis. Characterizing uncertainty in predictions of genomic sequence-to-activity models. In *Machine Learning in Computational Biology*, pp. 279–297. PMLR, 2024.

Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+, May 2020. URL http://arxiv.org/abs/1905.02928. arXiv:1905.02928 [stat].

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Jason D. Buenrostro, Beijing Wu, Howard Y. Chang, and William J. Greenleaf. ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. *Current Protocols in Molecular Biology*, 109(1), January 2015. ISSN 1934-3639, 1934-3647. doi: 10.1002/0471142727.mb2129s109. URL https://currentprotocols.onlinelibrary.wiley.com/doi/10.1002/0471142727.mb2129s109.

Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022.

Critical Assessment of Genome Interpretation Consortium. CAGI, the Critical Assessment of Genome Interpretation, establishes progress and prospects for computational genetic variant interpretation methods. *Genome Biol*, 25(1):53, February 2024. ISSN 1474-760X. doi: 10.1186/s13059-023-03113-6.

Bernardo P. de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark. DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. *Nat Genet*, 54(5):613–624, May 2022. ISSN 1546-1718. doi: 10.1038/s41588-022-01048-5. URL https://www.nature.com/articles/s41588-022-01048-5. Publisher: Nature Publishing Group.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pp. 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45014-6.

Kseniia Dudnyk, Donghong Cai, Chenlai Shi, Jian Xu, and Jian Zhou. Sequence basis of transcription initiation in the human genome. *Science*, 384(6694):eadj0116, 2024.

Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, October 2016. URL http://arxiv.org/abs/1506.02142. arXiv:1506.02142 [cs, stat].

Prashnna K. Gyawali, Xiaoxia Liu, James Zou, and Zihuai He. Ensembling improves stability and power of feature selection for deep learning models, October 2022. URL http://arxiv.org/abs/2210.00604. arXiv:2210.00604 [cs].

Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. Which Explanation Should I Choose? A Function Approximation Perspective to Characterizing Post Hoc Explanations. *Advances in Neural Information Processing Systems*, 35:5256–5268, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/22b111819c74453837899689166c4cf9-Abstract-Conference.html.

Adam Y He and Charles G Danko. Dissection of core promoter syntax through single nucleotide resolution modeling of transcription initiation. *bioRxiv*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Johannes C. Hingerl, Laura D. Martens, Alexander Karollus, Trevor Manz, Jason D. Buenrostro, Fabian J. Theis, and Julien Gagneur. scooby: Modeling multi-modal genomic profiles from DNA sequence at single-cell resolution, September 2024. URL https://www.biorxiv.org/content/10.1101/2024.09.19.613754v1. Pages: 2024.09.19.613754 Section: New Results.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [cs, stat].

Lara Hoffmann, Ines Fortmeier, and Clemens Elster. Uncertainty quantification by ensemble learning for computational optical form measurements. *Machine Learning: Science and Technology*, 2(3):035030, 2021.

Connie Huang, Richard W. Shuai, Parth Baokar, Ryan Chung, Ruchir Rastogi, Pooja Kathail, and Nilah M. Ioannidis. Personal transcriptome variation is poorly explained by current genomic deep learning models. *Nat Genet*, 55(12):2056–2059, 2023. ISSN 1061-4036. doi: 10.1038/s41588-023-01574-w. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10703684/.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Alireza Karbalayghareh, Merve Sahin, and Christina S. Leslie. Chromatin interaction–aware gene regulatory modeling with graph attention networks. *Genome Res*, 32(5):930–944, May 2022. ISSN 1088-9051. doi: 10.1101/gr.275870.121. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9104700/.

Alexander Karollus, Thomas Mauermeier, and Julien Gagneur. Current sequence-based models capture gene expression determinants in promoters but mostly ignore distal enhancers. *Genome Biology*, 24(1):56, March 2023. ISSN 1474-760X. doi: 10.1186/s13059-023-02899-9. URL https://doi.org/10.1186/s13059-023-02899-9.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. URL https://arxiv.org/abs/1703.04977.

W. J. Kent, A. S. Zweig, G. Barber, A. S. Hinrichs, and D. Karolchik. BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*, 26(17):2204–2207, July 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq351. URL https://doi.org/10.1093/bioinformatics/btq351. _eprint: https://academic.oup.com/bioinformatics/article-pdf/26/17/2204/48855290/bioinformatics_26_17_2204.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

Peter K Koo and Matt Ploenzke. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *Nature machine intelligence*, 3(3):258–266, 2021.

Peter K. Koo, Antonio Majdandzic, Matthew Ploenzke, Praveen Anand, and Steffan B. Paul. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLOS Computational Biology*, 17(5):e1008925, May 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008925. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008925. Publisher: Public Library of Science.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, November 2017. URL http://arxiv.org/abs/1612.01474. arXiv:1612.01474 [cs, stat].

Avantika Lal, Alexander Karollus, Laura Gunsalus, David Garfield, Surag Nair, Alex M Tseng, M Grace Gordon, Jenna L Collier, Nathaniel Diamant, Tommaso Biancalani, et al. Decoding sequence determinants of gene expression in diverse cellular and disease states. *bioRxiv*, pp. 2024–10, 2024.

Nicholas Keone Lee, Ziqi Tang, Shushan Toneyan, and Peter K. Koo. EvoAug: improving generalization and interpretability of genomic deep neural networks with evolution-inspired data augmentations. *Genome Biology*, 24(1):105, May 2023. ISSN 1474-760X. doi: 10.1186/s13059-023-02941-w. URL https://doi.org/10.1186/s13059-023-02941-w.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R Kelley. Predicting rna-seq coverage from dna sequence as a unifying model of gene regulation. *bioRxiv*, pp. 2023–08, 2023a.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R. Kelley. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation, September 2023b. URL https://www.biorxiv.org/content/10.1101/2023.08.30.555582v1. Pages: 2023.08.30.555582 Section: New Results.

Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions, November 2017. URL http://arxiv.org/abs/1705.07874. arXiv:1705.07874 [cs, stat].

Antonio Majdandzic, Chandana Rajesh, Ziqi Tang, Shushan Toneyan, Ethan L Labelson, Rohit K Tripathy, and Peter K Koo. Selecting deep neural networks that yield consistent attribution-based interpretations for genomics. In *Machine Learning in Computational Biology*, pp. 131–149. PMLR, 2022.

Antonio Majdandzic, Chandana Rajesh, and Peter K. Koo. Correcting gradient-based interpretations of deep neural networks for genomics. *Genome Biology*, 24(1):109, May 2023. ISSN 1474-760X. doi: 10.1186/s13059-023-02956-3. URL https://doi.org/10.1186/s13059-023-02956-3.

Stephen Malina, Daniel Cizin, and David A Knowles. Deep mendelian randomization: Investigating the causal knowledge of genomic deep learning models. *PLoS computational biology*, 18(10): e1009880, 2022.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble Distribution Distillation, November 2019. URL http://arxiv.org/abs/1905.00076. arXiv:1905.00076 [cs, stat].

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pp. 55–60. IEEE, 1994.

Gherman Novakovsky, Nick Dexter, Maxwell W Libbrecht, Wyeth W Wasserman, and Sara Mostafavi. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, 24(2):125–137, 2023.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive Confidence Machines for Regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen (eds.), *Machine Learning: ECML 2002*, pp. 345–356, Berlin, Heidelberg, 2002. Springer. ISBN 978-3-540-36755-0. doi: 10.1007/3-540-36755-1_29.

Alexander Sasse, Bernard Ng, Anna E. Spiro, Shinya Tasaki, David A. Bennett, Christopher Gaiteri, Philip L. De Jager, Maria Chikina, and Sara Mostafavi. Benchmarking of deep neural networks for predicting personal gene expression from DNA sequence highlights shortcomings. *Nat Genet*, 55(12):2060–2064, December 2023. ISSN 1546-1718. doi: 10.1038/s41588-023-01524-6. URL https://www.nature.com/articles/s41588-023-01524-6. Publisher: Nature Publishing Group.

Evan E Seitz, David M McCandlish, Justin B Kinney, and Peter K Koo. Interpreting cis-regulatory mechanisms from genomic deep neural networks using surrogate models. *bioRxiv*, pp. 2023.11.14.567120, March 2024. doi: 10.1101/2023.11.14.567120. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10680760/.

Dustin Shigaki, Orit Adato, Aashish N. Adhikari, Shengcheng Dong, Alex Hawkins-Hooker, Fumitaka Inoue, Tamar Juven-Gershon, Henry Kenlay, Beth Martin, Ayoti Patra, Dmitry D. Penzar, Max Schubach, Chenling Xiong, Zhongxia Yan, Alan P. Boyle, Anat Kreimer, Ivan V. Kulakovskiy, John Reid, Ron Unger, Nir Yosef, Jay Shendure, Nadav Ahituv, Martin Kircher, and Michael A. Beer. Integration of multiple epigenomic marks improves prediction of variant impact in saturation mutagenesis reporter assay. *Hum Mutat*, 40(9):1280–1291, September 2019. ISSN 1098-1004. doi: 10.1002/humu.23797.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000. ISSN 0378-3758. doi: 10.1016/S0378-3758(00)00115-4. URL https://www.sciencedirect.com/science/article/pii/S0378375800001154.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, December 2013. URL https://arxiv.org/abs/1312.6034v2.

Ziqi Tang, Nirali Somia, Yiyang Yu, and Peter K Koo. Evaluating the representational power of pre-trained dna language models for regulatory genomics. *bioRxiv*, pp. 2024–02, 2024.

The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, September 2012. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature11247. URL https://www.nature.com/articles/nature11247.

Shushan Toneyan, Ziqi Tang, and Peter K. Koo. Evaluating deep learning for predicting epigenomic profiles. *Nat Mach Intell*, 4(12):1088–1100, December 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00570-9. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10270674/.

Bindya Venkatesh and Jayaraman J. Thiagarajan. Heteroscedastic Calibration of Uncertainty Estimators in Deep Learning, October 2019. URL http://arxiv.org/abs/1910.14179. arXiv:1910.14179 [cs, stat].

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. Algorithmic Learning in a Random World. In *Algorithmic Learning in a Random World*. January 2005. doi: 10.1007/b106715. Journal Abbreviation: Algorithmic Learning in a Random World.

Zifan Wang, Haofan Wang, Shakul Ramkumar, Piotr Mardziel, Matt Fredrikson, and Anupam Datta. Smoothed Geometry for Robust Attribution. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13623–13634. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/9d94c8981a48d12adfeecfe1ae6e0ec1-Abstract.html.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated Residual Networks, 2017. URL https://arxiv.org/abs/1705.09914. Version Number: 1.

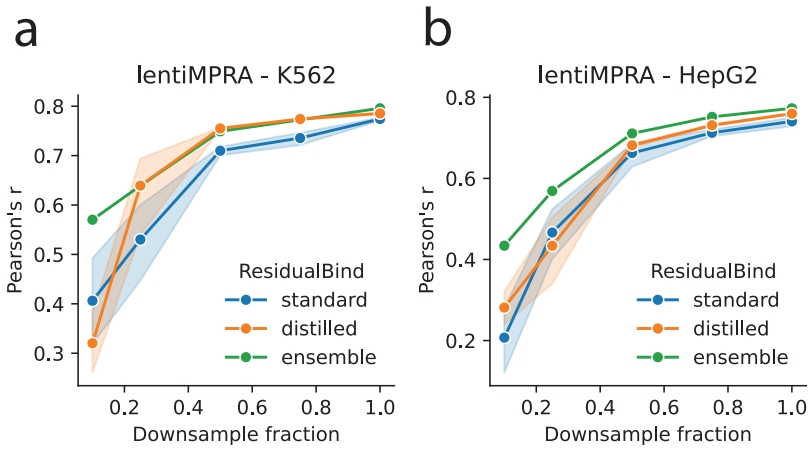
Yiyang Yu, Shivani Muthukumar, and Peter K Koo. Evoaug-tf: Extending evolution-inspired data augmentations for genomic deep learning to tensorflow. *Bioinformatics*, 40(3):btae092, 2024.

Daniel R. Zerbino, Nathan Johnson, Thomas Juettemann, Steven P. Wilder, and Paul Flicek. WiggleTools: parallel processing of large collections of genome-wide datasets for visualization and statistical analysis. *Bioinformatics*, 30(7):1008–1009, December 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt737. URL https://doi.org/10.1093/bioinformatics/btt737. _eprint: https://academic.oup.com/bioinformatics/article-pdf/30/7/1008/48921616/bioinformatics_30_7_1008.pdf.

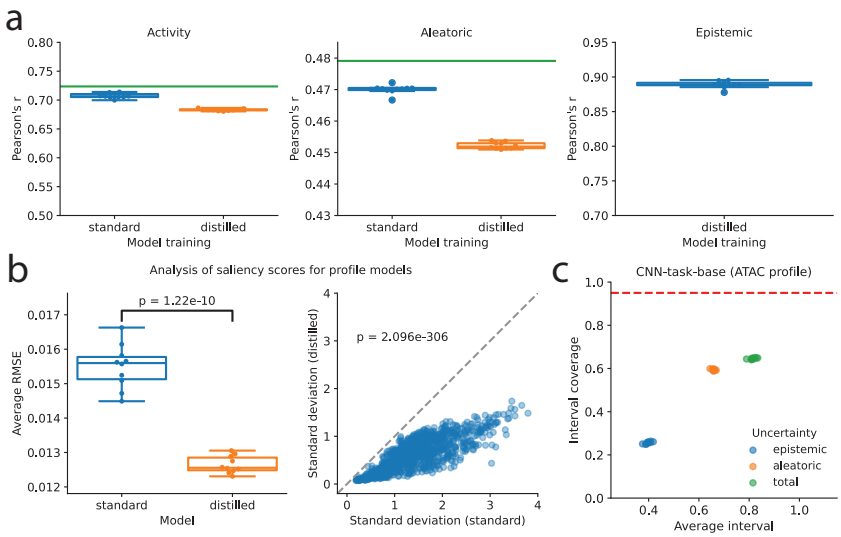
Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2022. doi: 10.1109/TPAMI.2021.3067100.

Jian Zhou. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. *Nat Genet*, 54(5):725–734, May 2022. ISSN 1546-1718. doi: 10.1038/s41588-022-01065-4.

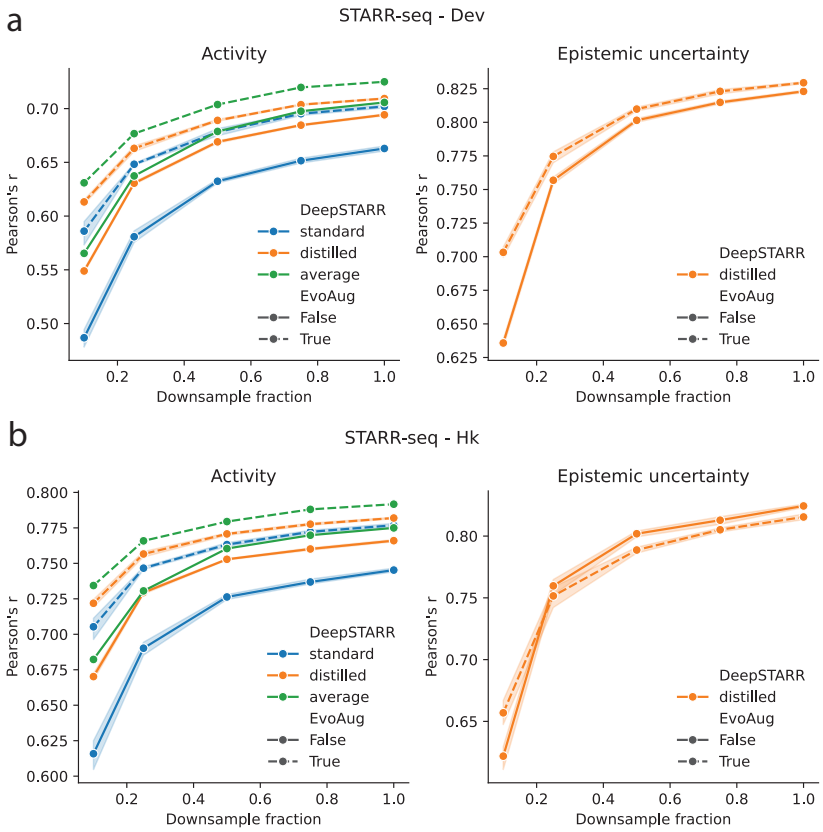
A Appendix Figures



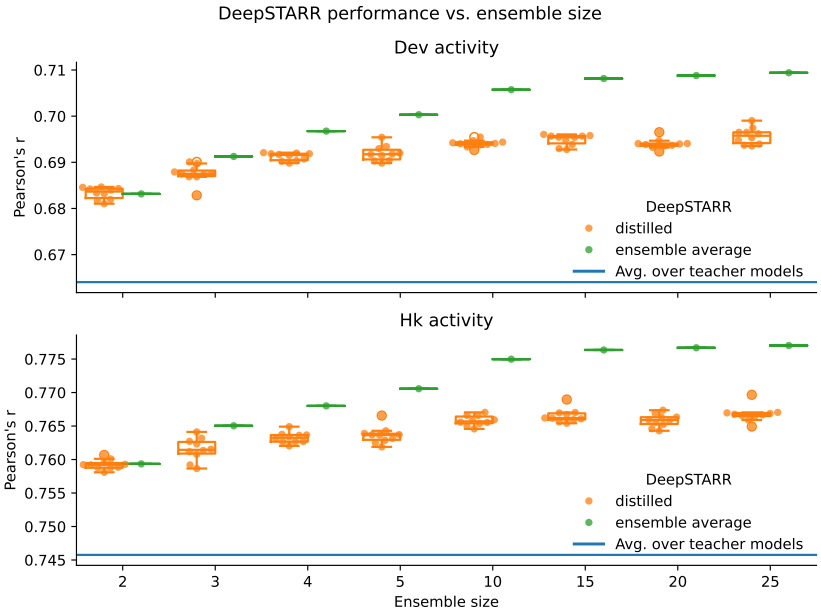
Appendix Figure 1: Comparing DEGU performance against benchmarks. (**a**,**b**) Performance of ResidualBind models trained on subsets of randomly downsampled lentiMPRA data for(**c**) K562 and (**d**) HepG2 cell lines. (**a**-**d**) Plots reflect teacher models with standard training (blue; $n = 10$), DEGU-distillation (orange; $n = 10$); and the ensemble average predictions (green). Shaded regions indicate 95% confidence intervals. Red dashed lines in (**c**,**d**) indicate performance of MPRAnn trained on full lentiMPRA dataset.



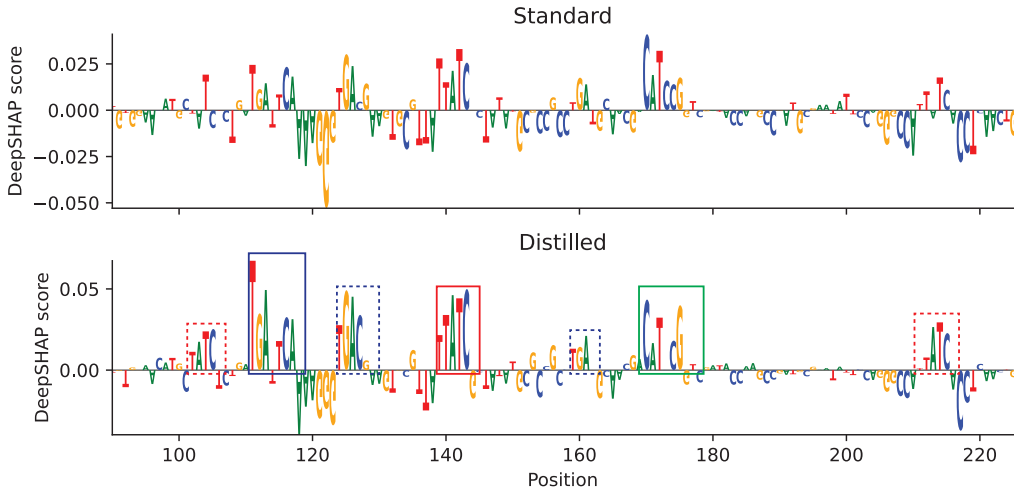
Appendix Figure 2: Performance on ATAC-seq profiles. (**a**) Boxplots of prediction performance of base-resolution CNN model trained on ATAC-seq profiles with standard training (blue) and DEGU distillation (orange) for the activity head (left), aleatoric head (middle) and epistemic head (right). Green horizontal line indicates the teacher ensemble performance. Boxplots represent $n = 10$ models trained with different random initializations, with the boxes indicating the first and third quartiles, the central line indicates the median, and whiskers denote the data range. (**b**) (left) Boxplot of average root mean squared error (RMSE) between attribution maps generated by individual models trained with standard training (blue) and DEGU distillation (orange), compared to the average attribution map across the teacher ensemble, and (right) scatter plot comparing the standard deviation of attribution scores across individual models trained with standard training ($n = 10$) and DEGU distillation ($n = 10$). Attribution scores were calculated with Saliency Maps for the activity output. RMSE is calculated for 1,000 high-activity test sequences. P-values indicate independent two-sided t-tests for average RMSE and paired two-sided t-tests for standard deviation. (**a**,**b**)Boxplots represent $n = 10$ models trained with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range. (**c**) Scatter plot of prediction interval coverage probability versus average interval size for epistemic uncertainty (blue) and aleatoric uncertainty (orange) and total uncertainty (green). Red dashed line indicates calibration with a 95% interval coverage probability. Each uncertainty quantification method is represented by $n = 10$ dots, indicating a model with different initializations, except for deep ensemble ($n = 1$).



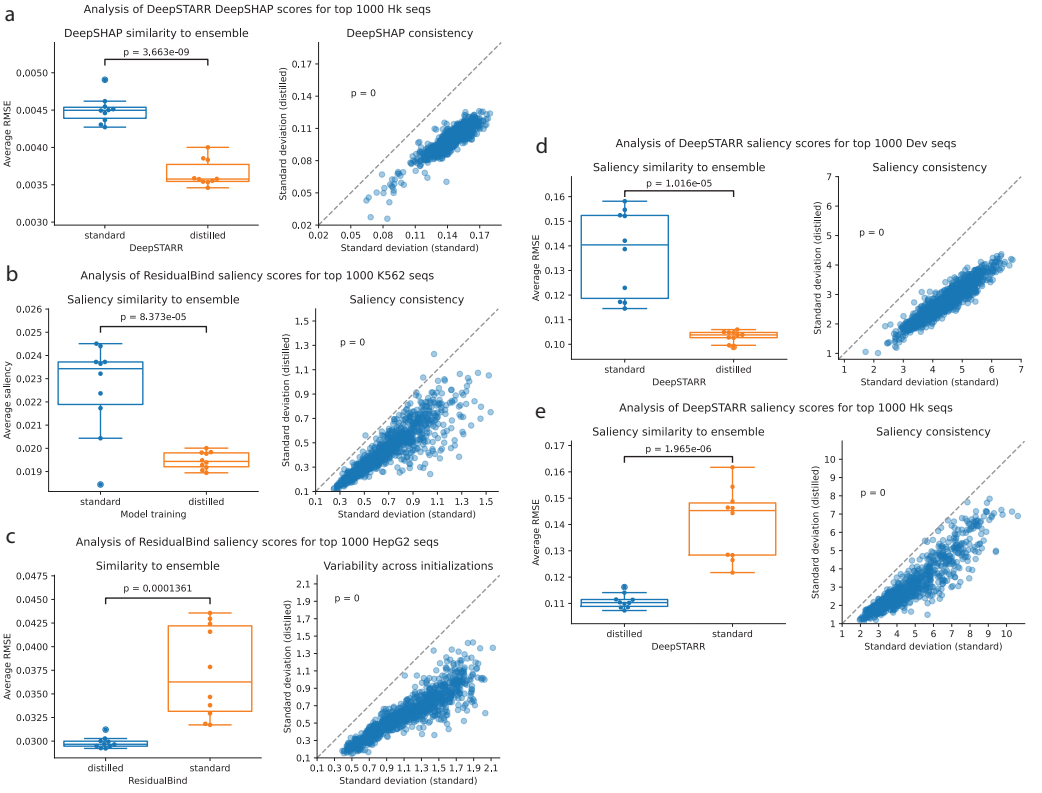
Appendix Figure 3: Comparison of model performance with downsampled training data using improved teacher ensembles. (**a,b**) Predictive performance for activity (left) and epistemic uncertainty (right) output heads of DeepSTARR models trained on different subsets of randomly downsampled STARR-seq data for (**a**) developmental (Dev) and (**b**) housekeeping (Hk) promoters. Dashed lines indicate DeepSTARR models trained with EvoAug data augmentations, while solid lines represent DeepSTARR models without data augmentations during training. Markers represent the average across 10 models with different random initializations and shaded region indicates 95% confidence interval.



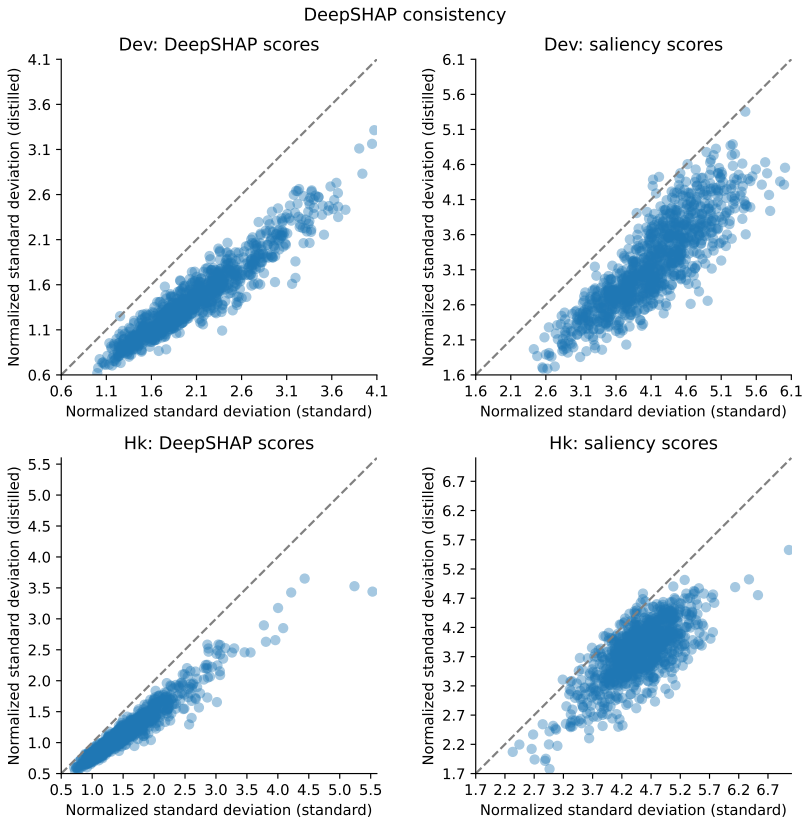
Appendix Figure 4: Performance comparison for different teacher ensemble sizes. Boxplot comparison of the performance of DeepSTARR ensembles (green) and distilled DeepSTARR models (orange) for different teacher ensembles comprised of varying number of models. The horizontal blue line represents the average performance metric (Pearson's r) of DeepSTARR with standard training. Boxplots represent $n = 10$ models trained with different random initializations, with the boxes indicating the first and third quartiles, the central line indicates the median, and whiskers denote the data range. Increasing the size of the teacher ensemble yields improvements in predictive accuracy for the ensemble average, but the predictive accuracy of distilled models saturates at a teacher ensemble size of $n = 10$.



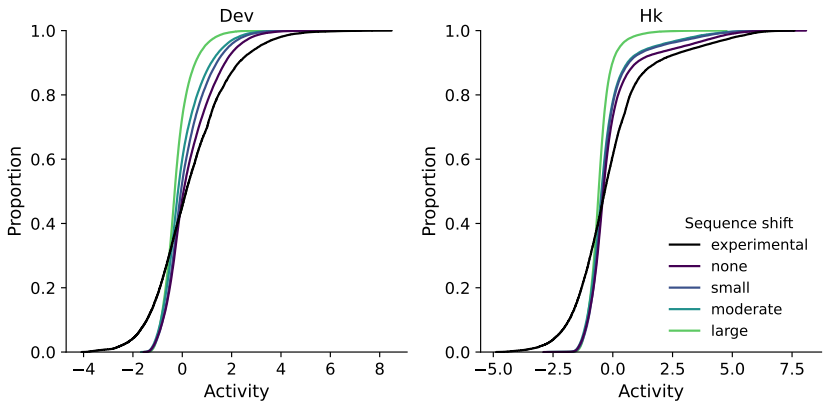
Appendix Figure 5: Attribution map for the Dev activity output head of an individual DeepSTARR model with standard training (top) and a DEGU-distilled DeepSTARR model (bottom) for an exemplary test sequence. Annotated boxes indicate binding sites for AP-1 (blue), GATA (red), and ETS/Twist (green), with solid lines indicating a strong match and dashed lines indicating a weak match.



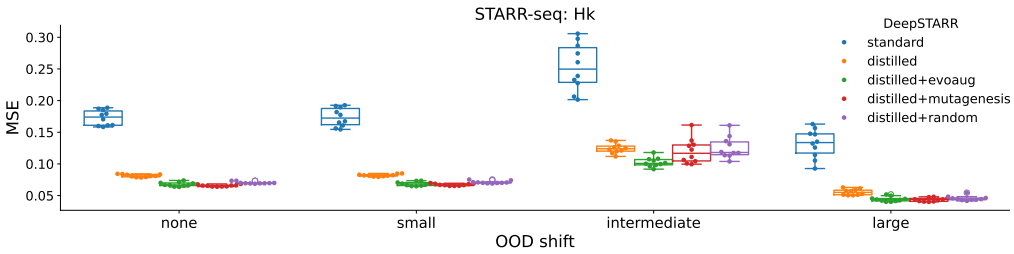
Appendix Figure 6: Additional attribution analysis performance comparisons. (left) Boxplots of average root mean squared error (RMSE) between attribution maps generated by individual models trained with standard training (blue) and DEGU distillation (orange), compared to the average attribution map across the teacher ensemble, and (right) scatterplots comparing the standard deviation of attribution scores across individual models trained with standard training ($n = 10$) and DEGU distillation ($n = 10$). Attribution scores were calculated with: (**a**) DeepSHAP for the Hk activity output head of DeepSTARR, (**b**,**c**) Saliency Maps for the activity output heads of ResidualBind models trained on (**b**) K562 and (**c**) HepG2 lentiMPRA data, (**d**, **e**) Saliency Maps for the (**d**) Dev and (**e**) Hk activity output heads of DeepSTARR. RMSE is calculated for 1,000 high-activity test sequences. P-values indicate independent two-sided t-tests for average RMSE and paired two-sided t-tests for standard deviation. Boxplots represent $n = 10$ models trained with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range.



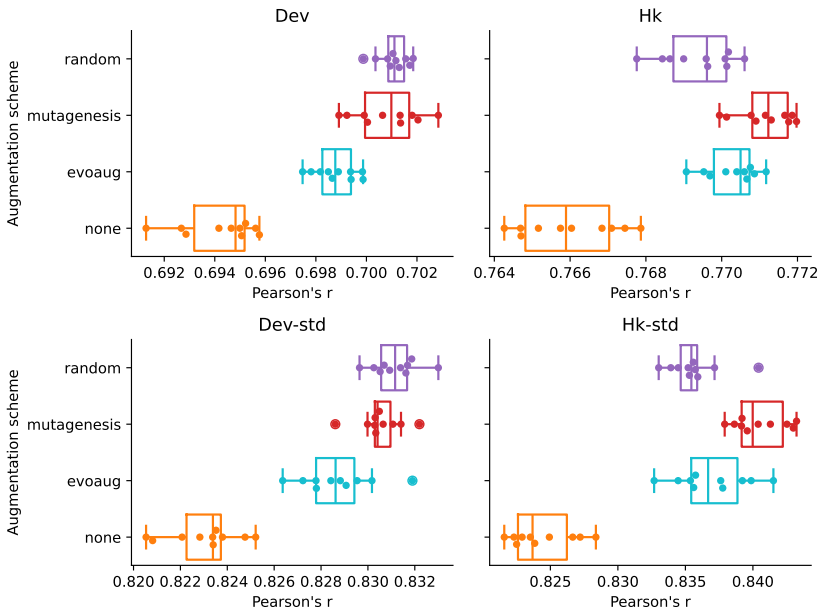
Appendix Figure 7: Magnitude-normalized attribution map consistency comparison. Scatter plots comparing the standard deviation of magnitude-normalized attribution scores across individual models trained with standard training ($n = 10$) and DEGU distillation ($n = 10$) for DeepSHAP applied (left) and Saliency Maps (right) applied to DeepSHAP for developmental promoters (top) and housekeeping promoters (bottom). Dots represent $n = 1,000$ high-activity test sequences.



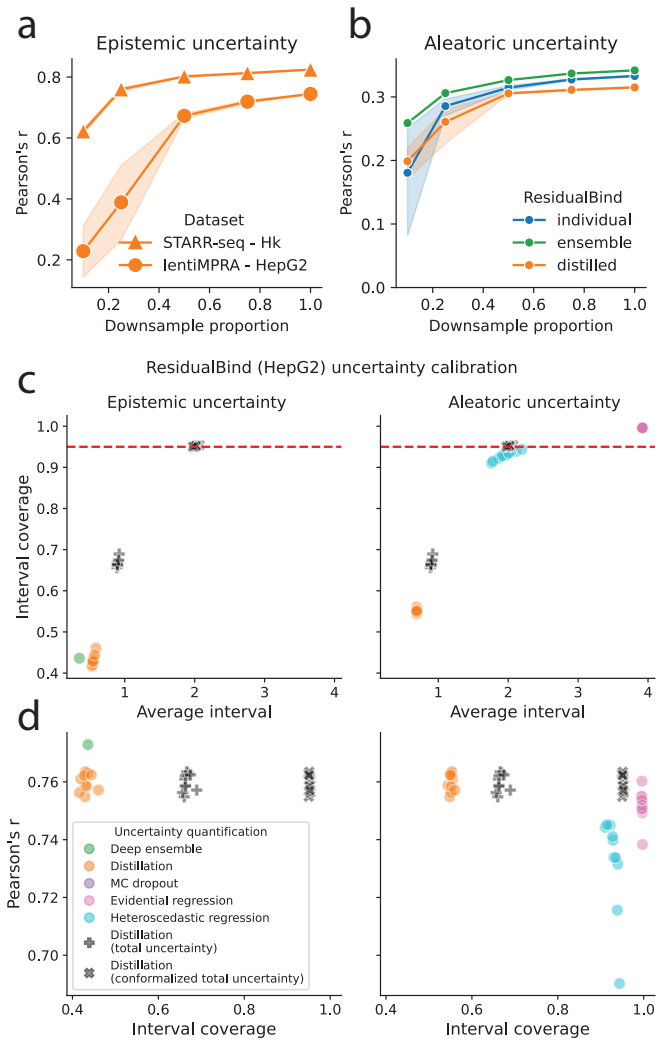
Appendix Figure 8: Distribution of predicted activity for STARR-seq test sequences under different distribution shifts. Cumulative distribution plot of predictions given by an exemplary distilled DeepSTARR model for sequences with different degrees of distribution shift: none (original test sequences), small (random mutagenesis with a rate of 5%), moderate (EvoAug mutagenesis), and large (random shuffling).



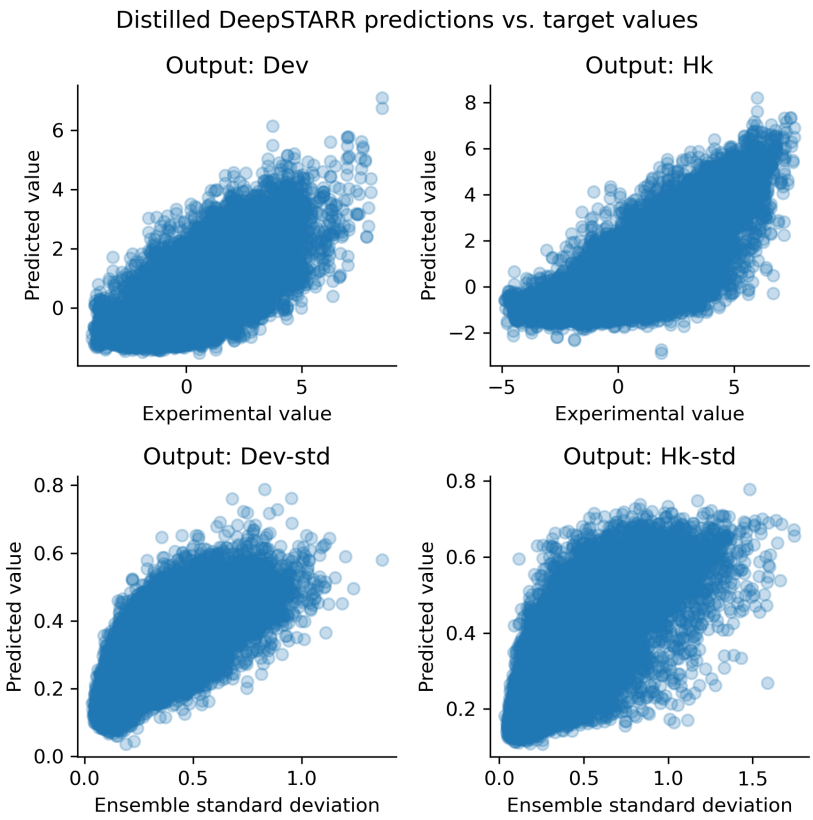
Appendix Figure 9: MSE between the teacher ensemble (DeepSTARR trained with EvoAug) and individual DeepSTARR models for Hk promoter activity across different training procedures: standard training (blue), DEGU-distillation (orange), and DEGU-distillation with dynamic EvoAug mutagenesis (green), partial random mutagenesis (red), and randomly shuffled sequences (purple). Results shown for sequences with varying degrees of distribution shift: none (original test set), small (partial random mutagenesis), intermediate (EvoAug mutagenesis), and large (random shuffle). Boxplots represent $n = 10$ models trained with different random initializations, with the boxes indicating the first and third quartiles, the central line indicating the median, and whiskers denote the data range.



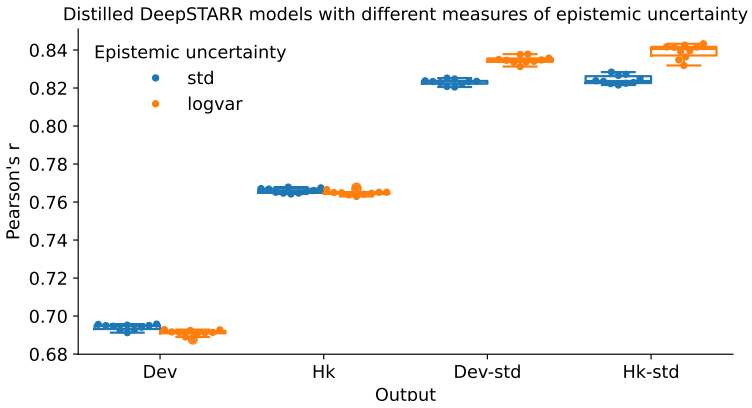
Appendix Figure 10: Performance of distilled DeepSTARR models trained with dynamic augmentations. Boxplots of predictive performance of activity head (top) and epistemic uncertainty head (bottom) from distilled DeepSTARR models trained with different dynamic sequence augmentations for Dev (left) and Hk promoters (right). Boxplots represent $n = 10$ models trained with different random initializations, with the boxes indicating the first and third quartiles, the central line indicates the median, and whiskers denote the data range.



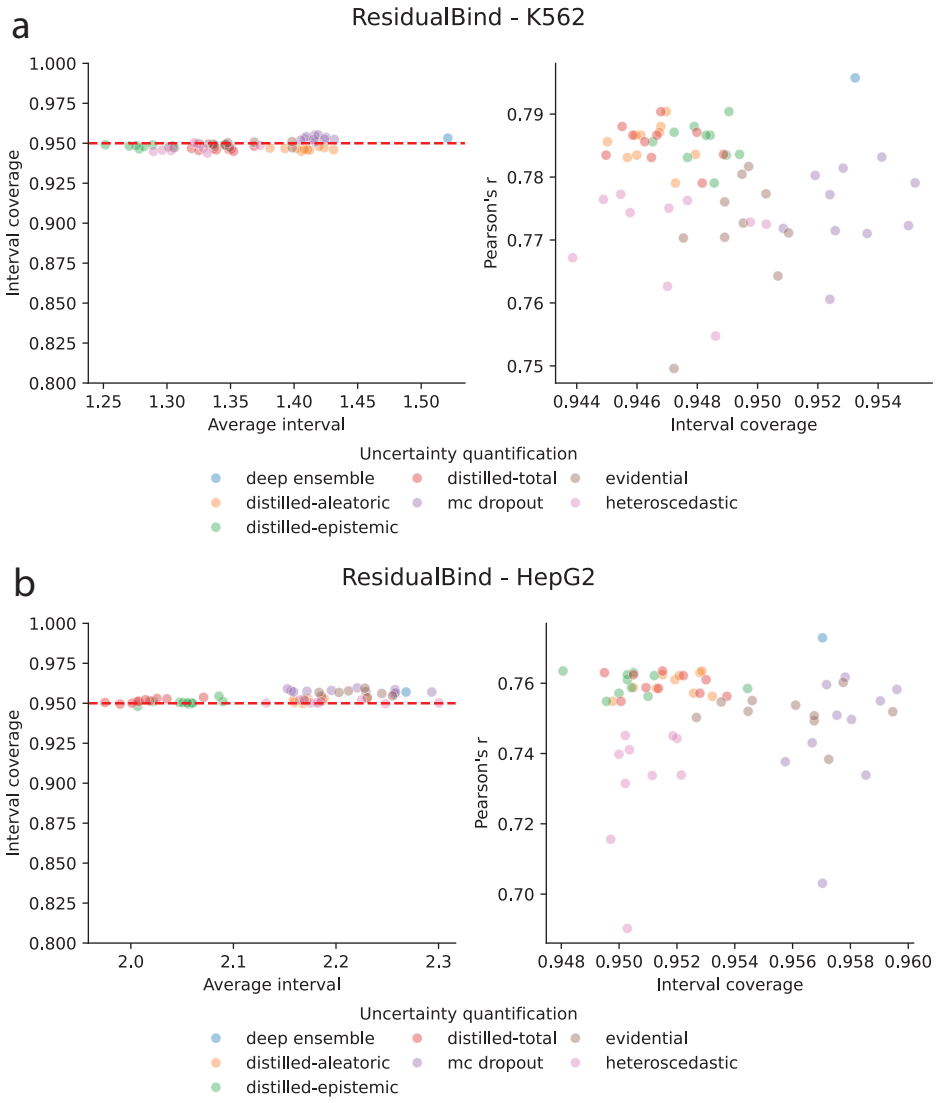
Appendix Figure 11: Additional performance comparison of uncertainty estimates. (**a**,**b**) Predictive performance for (**a**) epistemic and (**b**) aleatoric uncertainty for models with standard training (blue), DEGU-distillation (orange), and the teacher ensemble (green), trained on subsets of randomly downsampled training data. Markers represent the average across $n = 10$ models with different random initializations and shaded region indicates 95% confidence interval. Results are shown for (**a**) the Hk epistemic uncertainty head of distilled DeepSTARR models and the epistemic uncertainty output head of distilled ResidualBind models trained on HepG2 lentiMPRA data, and (**b**) the aleatoric uncertainty output heads of ResidualBind models trained on HepG2 lentiMPRA data. (**c**,**d**) Scatter plots of (**c**) prediction interval coverage probability and (**d**) predictive accuracy versus average interval size for different uncertainty quantification methods for epistemic uncertainty (left) and aleatoric uncertainty (right). Uncertainty quantification methods are based on ResidualBind model trained on HepG2 lentiMPRA data. Red dashed line indicates calibration with a 95% interval coverage probability. Each uncertainty quantification method is represented by $n = 10$ dots, indicating a model with different initializations, except for deep ensemble ($n = 1$).



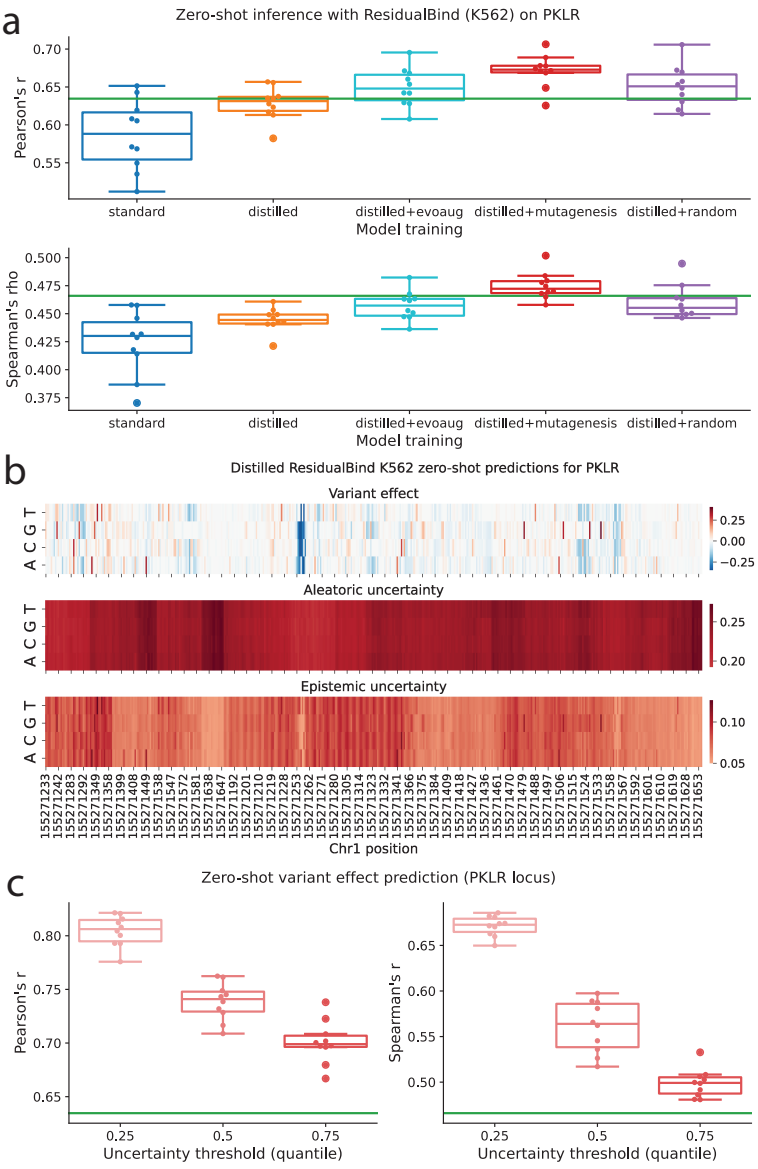
Appendix Figure 12: Prediction comparison for distilled DeepSTARR. Scatter plots comparing distilled DeepSTARR model predictions and target values for enhancer activity (top row) and epistemic uncertainty (bottom row) for developmental (left) and housekeeping (right) promoters. Each dot represents a different test sequence ($n = 41186$)



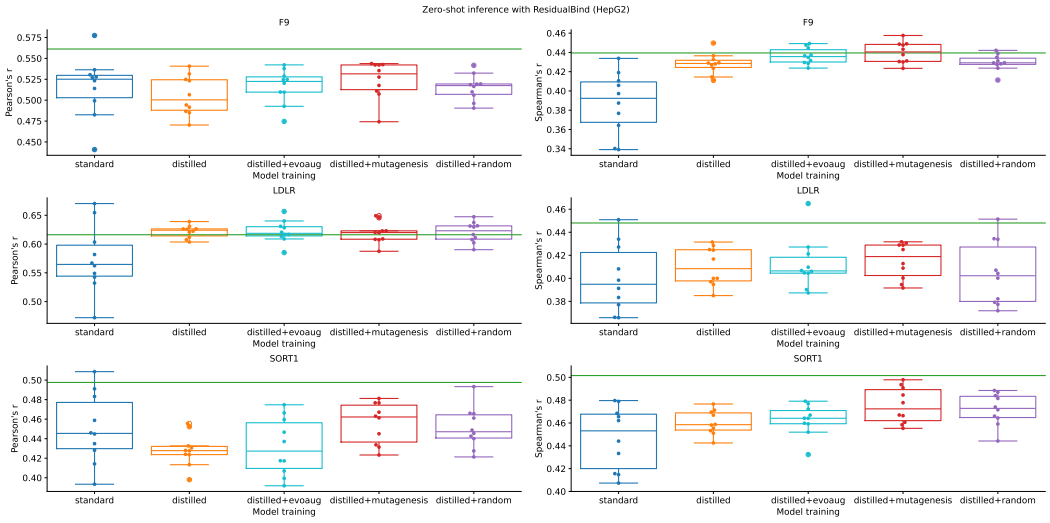
Appendix Figure 13: Comparison of different measures of variability. Boxplots of predictive performance for the epistemic uncertainty output head of different distilled DeepSTARR models where epistemic uncertainty ea trained on standard deviation (blue) versus log variance (orange) of the predictions across the teacher ensemble. Boxplots represent $n = 10$ models trained with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range.



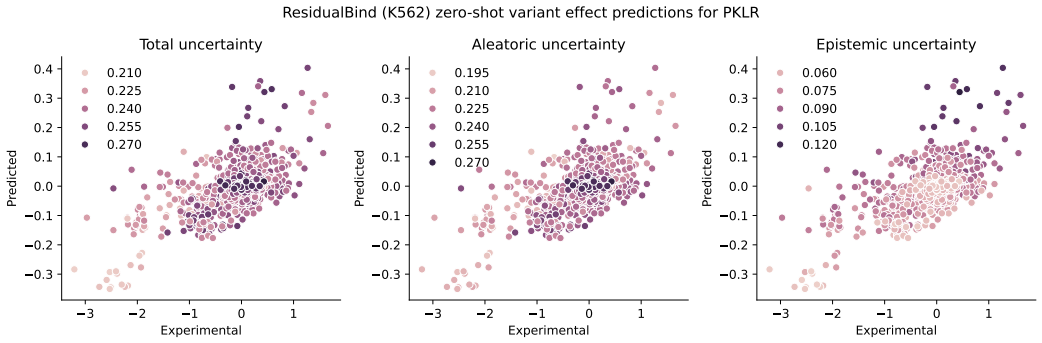
Appendix Figure 14: Comparison of loss functions for training ResidualBind. Boxplots comparing performance of ResidualBind trained on lentiMPRA data for K562 (top) and Hepg2 (bottom) using MSE loss or heteroscedastic loss (columns). Models trained with MSE loss learn aleatoric uncertainty from replicate level variation while models trained with heteroscedastic regression implicitly learn aleatoric uncertainty during the training process (but replicate level variation is used to calculate Pearson's r). Boxplots represent $n = 10$ models with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range.



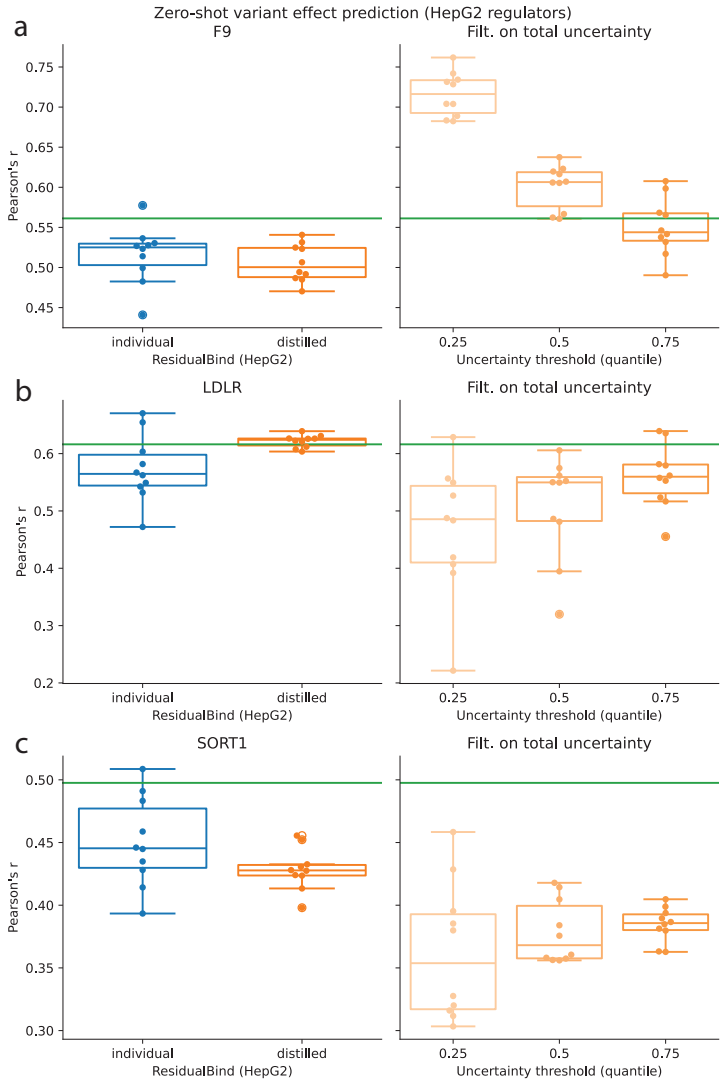
Appendix Figure 15: Single-nucleotide variant effect generalization. (**a**) Boxplots of zero-shot variant effect predictive performance for models with standard training, DEGU-distillation, and DEGU-distillation with dynamic augmentations. Predictive performance was evaluated using Pearson's r (top) and Spearman's rho (bottom). (**b**) Boxplots of zero-shot variant effect predictive performance for models with standard training (blue) and DEGU-distillation with random mutagenesis augmentations (red) for variants filtered on different quantile thresholds of predicted total uncertainty according to different correlation metrics: Pearson's r (left) and Spearman's rho (right). (**a**,**b**) Green horizontal line indicates the performance of the teacher ensemble. Boxplots represent $n = 10$ ResidualBind models trained on K562 lentiMPRA data with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range.(**c**) Heat map showing predicted effect size (top), aleatoric uncertainty (middle), and epistemic uncertainty (bottom) given by a representative distilled ResidualBind model trained on K562 lentiMPRA data for all CAGI variants provided for the *PKLR* locus.



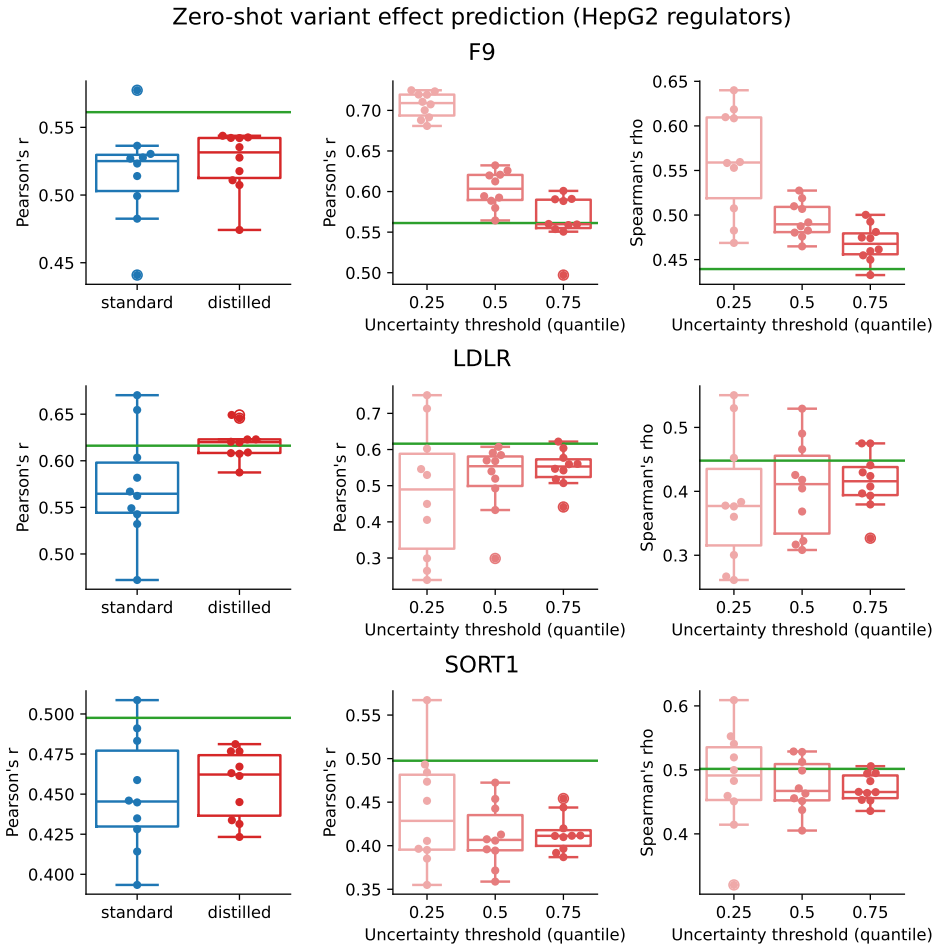
Appendix Figure 16: Single nucleotide variant effect prediction for HepG2 regulators. Boxplots of zero-shot variant effect predictive performance for models with standard training (blue); DEGU-distillation (orange); and DEGU-distillation with dynamic augmentations. Boxplots represent $n = 10$ models with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range. Green horizontal line indicates the performance of the teacher ensemble.



Appendix Figure 17: Single-nucleotide variant effect prediction performance with uncertainty annotations. Scatter plots of predictive performance of ResidualBind trained with random mutagenesis augmentations and single-nucleotide variant effects in the PKLR locus measured via an MPRA in K562. The color of each dot represents the uncertainty according to total uncertainty (left), aleatoric uncertainty (middle), and epistemic uncertainty (right).



Appendix Figure 18: Conformal prediction analysis. Scatter plots of prediction interval coverage probability (left) and predictive accuracy (right) versus average interval size for different uncertainty quantification methods after conformalizing estimates on validation data. The results are shown for ResidualBind models trained on (**a**) K562 and (**b**) HepG2 lentiMPRA data. (left) Red dashed line indicates 95% interval. Each uncertainty quantification method is represented by $n = 10$ dots, indicating a model with different initializations, except for deep ensemble ($n = 1$).



Appendix Figure 19: Additional analysis for single-nucleotide variant effect generalization for HepG2 regulators. Boxplots of zero-shot variant effect predictive performance for models with standard training (blue) and DEGU-distillation with mutagenesis augmentations (red) for all nucleotide variants (left) and for variants filtered on different quantile thresholds of predicted total uncertainty (middle, right). Boxplots represent $n = 10$ models with different random initializations, with the boxes representing the first and third quartiles, the central line indicating the median, and whiskers denoting the data range. Green horizontal line indicates the performance of the teacher ensemble.

B METHODS

DATASETS

FLY ENHANCER ACTIVITY WITH STARR-SEQ

We obtained STARR-seq data for developmental (Dev) and housekeeping (Hk) promoters in *D. melanogaster* S2 cells from de Almeida et al.de Almeida et al. (2022) Each sequence is 249 base pairs (bp) long. Enhancer activity for both housekeeping and developmental classes was predicted simultaneously as a multi-task regression. The data was split into train, test, and validation sets containing 402296, 41186, and 40570 samples, respectively.

HUMAN REGULATORY SEQUENCES WITH LENTIMPRA

We used lentiMPRA data for K562 and HepG2 cell lines from Agarwal et al.Agarwal et al. (2023). Each 230 bp cis-regulatory sequence was associated with a scalar activity measurement for three biological replicates. The mean and standard deviation across the replicates was used as target values for regulatory sequence activity and aleatoric uncertainty, respectively. For each cell type, we performed two types of regressions: 1) a single-task regression for regulatory activity only, and 2) a multi-task regression for both regulatory activity and aleatoric uncertainty. We generated a different dataset for each regression task. For the single-task regression, we removed any samples for which an activity measurements was provided without corresponding sequence data was not available. For the multi-task regression, we also removed samples for which experimental data from at least two replicates was not available, due to the inability to calculate aleatoric uncertainty. For each dataset, we randomly split the training, validation, and test sets according to the fractions 0.8, 0.1, and 0.1, respectively, ensuring that any forward and reverse complement sequence pairs would be assigned to the same set to avoid data leakage. The HepG2 data for single-task regression (activity only) was split into train, test, and validation sets containing 111901, 13988, and 13988 samples, respectively. The HepG2 data for the multi-task regression (activity and aleatoric uncertainty) was split into train, test, and validation sets containing 111518, 13939, and 13942 samples, respectively. The K562 data for single-task regression (activity only) was split into train, test, and validation sets containing 181002, 22626, and 22626 samples, respectively. The K562 data for multi-task regression (activity and aleatoric uncertainty) was split into train, test, and validation sets containing 180564, 22571, and 22570 samples, respectively.

PROFILE-BASED CHROMATIN ACCESSIBILITY WITH GOPHER

We acquired hg38-aligned ATAC-seq bigWig files for A549 cells (ENCSR032RGS) from ENCODE The ENCODE Project Consortium (2012). We processed the 3 replicate fold change over control bigwig files into 2 bigWig files: 1) average read coverage across replicates and 2) standard deviation of read coverage across replicates. Wiggletools Zerbino et al. (2013) and UCSC's bedGraphToBigWig Kent et al. (2010) were used to wrangle the data into bigWig file formats. Following a previously published data processing procedure Toneyan et al. (2022), we divided each chromosome into equal, non-overlapping 3072 bp bins. We one-hot encoded each sequence with matched base-resolution coverage tracks from the average and standard deviation bigWig files. We split the dataset into a test set comprising chromosome 8, a validation set comprising chromosome 9, and a training set encompassing the remaining chromosomes, with the exclusion of chromosome Y and contigs. Performance was assessed as the Pearson correlation across the whole chromosome as outlined in Ref. Toneyan et al. (2022).

SINGLE-NUCLEOTIDE VARIANT EFFECT WITH CAGI5

The CAGI5 challenge dataset Critical Assessment of Genome Interpretation Consortium (2024); Shigaki et al. (2019), which consists of experimentally measured saturation mutagenesis of a 230 bp regulatory element via a MPRA, was used to evaluate the performance of the ResidualBind models on zero-shot single-nucleotide variant effect generalization. We considered only experiments in HepG2 (*LDLR*, *F9*, *SORT1*) and K562 (*PKLR*). We extracted 230 bp sequences from the reference genome (hg19) centered on each single-nucleotide variant in the CAGI data. We calculated the predicted effect of each allele as: $\hat{y}_{alt} - \hat{y}_{ref}$, where $\hat{y}_{alt}$ is the model's activity prediction for

the alternate allele and $\hat{y} - ref$ is the model's activity prediction for the corresponding reference allele. Performance was evaluated as the Pearson correlation between the predicted effect and the experimentally measured effect.

MODELS

DEEPSTARR FOR STARR-SEQ

We implemented DeepSTARR de Almeida et al. (2022) as described in Ref de Almeida et al. (2022), according to:

1. 1D convolution (256 kernels, size 7, batch normalization, ReLU activation)
   1D max-pooling (size 2)
2. 1D convolution (60 kernels, size 3, batch normalization, ReLU activation)
   1D max-pooling (size 2)
3. 1D convolution (60 kernels, size 5, batch normalization, ReLU activation)
   1D max-pooling (size 2)
4. 1D convolution (120 kernels, size 3, batch normalization, ReLU activation)
   1D max-pooling (size 2)
5. flatten
6. linear (256 units, batch normalization, ReLU activation)
   dropout(0.4)
7. linear (256 units, batch normalization, ReLU activation)
   dropout(0.4)
8. output (2 units, linear)

The 2 units in the output layer represent the Dev and Hk enhancer activities. For distilled models which predict both activity and epistemic uncertainty, the output layer is increased to 4 units, representing Dev activity, Hk activity, Dev epistemic uncertainty, and Hk epistemic uncertainty.

RESIDUALBIND FOR LENTIMPRA

We used a custom ResidualBind modelKoo et al. (2021); Tang et al. (2024), a CNN with dilated residual blocks He et al. (2016); Yu et al. (2017), to model lentiMPRA data. The ResidualBind architecture is as follows:

1. 1D convolution (196 kernels, size 19, batch normalization, SiLU activation)
   dropout (0.2)
2. Dilated residual block (5 dilations)
   1D convolution (196 kernels, size 3, batch normalization, ReLU activation)
   dropout (0.1)
   1D convolution (196 kernels, size 3, dilation rate 1, batch normalization, ReLU activation)
   dropout (0.1)
   1D convolution (196 kernels, size 3, dilation rate 2, batch normalization, ReLU activation)
   dropout (0.1)
   1D convolution (196 kernels, size 3, dilation rate 4, batch normalization, ReLU activation)
   dropout (0.1)
   1D convolution (196 kernels, size 3, dilation rate 8, batch normalization, ReLU activation)
   dropout (0.1)
   1D convolution (196 kernels, size 3, dilation rate 16, batch normalization, ReLU activation)
   dropout (0.1)

skip connection to input
SiLU activation
dropout (0.2)
1D max-pooling (size 5)

3. 1D convolution (256 kernels, size 7, batch normalization, SiLU activation)
   dropout (0.2)
   1D max-pooling (size 5)
4. linear (256 units, batch normalization, SiLU activation)
   dropout(0.5)
5. 1D global average pooling
6. flatten
7. linear (256 units, batch normalization, SiLU activation)
   dropout(0.5)
8. output (1 unit, linear)

For ResidualBind models trained on both the replicate average and standard deviation, the output layer is increased to 2 units representing activity and aleatoric uncertainty, respectively. For distilled ResidualBind models, the output layer is increased to 3 units representing activity, aleatoric uncertainty, and epistemic uncertainty.

CNN-TASK-BASE

A base-resolution CNN from Ref. Toneyan et al. (2022) was used to fit the ATAC-seq profile data. Briefly, CNN-task-base is composed of 3 convolutional blocks, which consist of a 1D convolution, batch normalization, activation, max pooling and dropout, followed by 2 fully-connected blocks, which includes a dense layer, batch normalization, activation, and dropout. The first fully connected block scales down the size of the representation, serving as a bottleneck layer. The second fully-connected block rescales the bottleneck to the target resolution. This is followed by another convolutional block. The representations from the outputs of the convolutional block is then input into task-specific output heads; each head consists of a convolutional block followed by a linear output layer with softplus activations. The activation of all hidden layers are ReLU.

TRAINING MODELS

STANDARD TRAINING OF DEEPSTARR AND RESIDUALBIND

We uniformly trained each model by minimizing the mean-squared error loss function with mini-batch stochastic gradient descent (100 sequences) for 100 epochs with Adam updates using default parameters Kingma & Ba (2017). The learning rate was initialized to 0.001 and was decayed by a factor of 0.1 when the validation loss did not improve for 5 epochs. All reported performance metrics are drawn from the test set using the model parameters from the epoch which yielded the lowest loss on the validation set. For each model, we trained 10 different individual models with different random initializations.

STANDARD TRAINING OF CNN-TASK-BASE

CNN-task-base models were trained using a Poisson loss and Adam with default parameters and a minibatch size of 100. The learning rate was initialized to 0.001 and was decayed by a factor of 0.3 when the validation loss did not improve for 5 epochs. All reported performance metrics are drawn from the test set using the model parameters from the epoch which yielded the lowest loss on the validation set. During training, random shift and stochastic reverse-complement data augmentations were used Toneyan et al. (2022). Random shift is a data augmentation that randomly translates the input sequence (and corresponding targets) online during training. For each mini-batch, a random sub-sequence of 2048 bp and its corresponding target profile was selected separately for each sequence. Reverse-complement data augmentation is also employed online during training. During each mini-batch, half of training sequences were randomly selected and replaced by their

reverse-complement sequence. For those sequences that were selected, the training target was correspondingly replaced by the reverse of original coverage distribution. For each model, we trained 10 different individual models with different random initializations.

TRAINING DEEPSTARR WITH EVOAUG

Models trained with EvoAug-TF Yu et al. (2024) use the following augmentation settings:

- random deletions with a size range of 0-20bp (applied per batch)
- random translocation with a size range of 0-20bp (applied per batch)
- random Gaussian noise with $\mu = 0$ and $\sigma = 0.2$ added to each variant in the input sequence (applied per sequence)
- random mutation of 5% of nucleotides in sequence (applied per sequence)

For each minibatch during training, one of the augmentations is randomly selected from the list of possible augmentations described above and applied to every sequence in the minibatch. Both teacher and student models were trained with the same optimizer, learning rate decay, and early stopping hyperparameters described for standard training.

DEGU: DISTILLING KNOWLEDGE OF ENSEMBLES TO UNCERTAINTY-AWARE GENOMIC DNNS

Ensemble Training. For each prediction task, we trained an ensemble of M models, each denoted as f_{θ_m}, where $m = 1, 2, \ldots, M$, with identical architectures but different random initializations, θ_m. Each model in the ensemble outputs a prediction $f_{\theta_m}(x_i)$ for a given input sequence $x_i \forall i = 1, 2, \ldots, N$.

The predictions across the ensemble for input x_i are aggregated to compute the ensemble mean μ_i and standard deviation σ_i, defined as:

$$\mu_i = \frac{1}{M} \sum_{m=1}^{M} f_{\theta_m}(x_i)$$

$$\sigma_i = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(f_{\theta_m}(x_i) - \mu_i \right)^2}$$

Here, μ_i represents the mean prediction (capturing the central tendency of the ensemble), while σ_i captures the epistemic uncertainty (i.e., the variability across the ensemble's predictions).

Distilled Model Training. To train the distilled model, we introduce multitask learning by incorporating the ensemble mean μ_i and standard deviation σ_i into the loss function. Let the distilled model be denoted by $g_\phi(x_i)$, parameterized by ϕ. The distilled model is trained with two output heads:

- A mean prediction head $g_\phi^{(\mu)}(x_i)$ that approximates the ensemble mean μ_i.
- An uncertainty prediction head $g_\phi^{(\sigma)}(x_i)$ that predicts the ensemble standard deviation σ_i, capturing the epistemic uncertainty.

Thus, the complete output of the distilled model for sequence x_i is given by:

$$g_\phi(x_i) = \left(g_\phi^{(\mu)}(x_i), g_\phi^{(\sigma)}(x_i) \right)$$

Loss Function. The training objective for the distilled model is defined as a multitask loss function that includes the error in predicting both the ensemble mean and the epistemic uncertainty. The loss function $\mathcal{L}$ can be written as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{L}_{\text{mean}}(g_\phi^{(\mu)}(x_i), \mu_i) + \lambda\mathcal{L}_{\text{uncertainty}}(g_\phi^{(\sigma)}(x_i), \sigma_i)\right)$$

Where:

- $\mathcal{L}_{\text{mean}}$ is the loss for the mean prediction, e.g., mean squared error (MSE):

$$\mathcal{L}_{\text{mean}} = \frac{1}{2}\left(g_\phi^{(\mu)}(x_i) - \mu_i\right)^2$$

- $\mathcal{L}_{\text{uncertainty}}$ is the loss for predicting the epistemic uncertainty, e.g., MSE:

$$\mathcal{L}_{\text{uncertainty}} = \frac{1}{2}\left(g_\phi^{(\sigma)}(x_i) - \sigma_i\right)^2$$

- λ is a hyperparameter that controls the weighting of the uncertainty prediction loss. In this study, we use $\lambda = 1$

Aleatoric Uncertainty (Optional). In cases where at least 3 experimental replicates are available, aleatoric uncertainty $\sigma_{\text{aleatoric}}$ can also be predicted. The aleatoric uncertainty is approximated by the variability across the experimental replicates and incorporated into the training of the student model through an additional output head. The loss function can be extended as:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \gamma\mathcal{L}_{\text{aleatoric}}(g_\phi^{(\text{aleatoric})}(x_i), \sigma_{\text{aleatoric}})$$

Where γ is a hyperparameter controlling the contribution of aleatoric uncertainty to the overall loss. In this study, we use $\gamma = 1$.

Model Architecture. While the distilled models can be comprised of any architecture, in this study, the distilled models share the same architecture as the original ensemble models, with the exception of the final layer. If the original models had N_{out} output heads, the distilled models will have $2N_{\text{out}}$ heads to account for both the mean and epistemic uncertainty predictions. In cases where aleatoric uncertainty is also modeled, the distilled models will have $3N_{\text{out}}$ output heads.

DISTILLING DEEPSTARR

We first trained an ensemble of 10 DeepSTARR models with different random initializations on STARR-seq data. We then used each individual model in the ensemble to make predictions on the training sequences and calculated the average and standard deviation across the 10 models. These values were used as target values for activity and epistemic uncertainty, respectively. Then, we trained 10 distilled models with different random initializations following the same procedure as standard training using the new target labels generated by the teacher ensemble.

DISTILLING RESIDUALBIND

For each cell type, we also trained an ensemble of 10 ResidualBind models with different random initializations on both the mean and standard deviation of experimental activity values across the biological replicates in the lentiMPRA data, with the latter value representing aleatoric uncertainty. The averages of the activity and aleatoric uncertainty predictions from this ensemble were used as new target values for training the distilled models. Moreover, the standard deviation of the activity predictions were also used to generate labels for the epistemic uncertainty as with DeepSTARR. Then, we trained 10 distilled models with different random initializations following the same procedure as standard training using the new target labels generated by the teacher ensemble.

DISTILLING CNN-TASK-BASE

We first trained an ensemble of 10 CNN-task-base models with different random initializations on profile-based ATAC-seq data. These models were trained to predict the mean and standard deviation

of the profiles across the three biological replicates. We then used the ensemble of models to generate new bigWig tracks based on the mean and standard deviation of the profiles across the 10 models, using wiggletools. We also included the mean of the aleatoric uncertainty head, resulting in 3 bigWig tracks all generated by the predictions of the ensemble. Each bigWig was processed following the same procedure as the original ATAC-seq profiles. Distilled models used the same architecture and training procedure as the standard CNN-task-base with the exception of using ensemble-generated labels and the addition of the epistemic profile prediction task, resulting in 3 prediction tasks, including the mean profile, the aleatoric uncertainty profile, and the epistemic uncertainty profile. Then, we trained 10 distilled models with different random initializations following the same procedure as standard training using the new target labels generated by the teacher ensemble.

DISTILLING MODELS WITH DYNAMIC AUGMENTATIONS

During distillation, we generated data augmentations following three different augmentation schemes:

1. EvoAug-TFYu et al. (2024), with n augmentations selected from the same augmentation list described above where n is randomly selected from 0 to 2.
2. Random mutagenesis using the EvoAug-TF implementation with a mutation fraction of 5%
3. Random shuffling

The augmentation is applied to each minibatch during training, replacing the original training sequences. The ensemble of teacher models is used to make predictions on the augmented sequences, and the average and standard deviation of these predictions are used as target values for activity and uncertainty, respectively. The original validation and test sequences were used for early stopping and evaluations. Student models were trained with the same optimizer, learning rate decay, and early stopping hyperparameters described for standard training.

OOD GENERALIZATION ANALYSIS

We generated out-of-distribution (OOD) sequences using the following sampling methods:

1. Small distribution shift: random mutagenesis with a mutation fraction of 5% generated with EvoAugLee et al. (2023).
2. Moderate distribution shift: evolution-inspired mutagenesis generated with 2 augmentations selected from the same augmentation list described above using EvoAugLee et al. (2023)
3. Large distribution shift: random shuffling of the test sequences.

Activity labels for these OOD sequences were obtained by averaging the predictions from an *in silico* oracle comprised of an ensemble of DeepSTARR models trained with EvoAug (with the same hyperparameters as stated above).

ATTRIBUTION ANALYSIS

Saliency Maps Simonyan et al. (2013) and DeepSHAP Lundberg & Lee (2017) scores were employed for attribution analysis to elucidate the input nucleotides most influential model predictions. For each sequence activity output head of each model, we generated attribution maps for 1000 sequences from the test set associated with the largest target values. Each method yielded a $4 \times L$ map where L is the length of the input sequence. For DeepSHAP, background sequences were comprised of 100 randomly selected sequences from the test set. Gradient correction was applied to all attribution maps by subtracting the average attribution score across all channels (nucleotides) at each position Majdandzic et al. (2023). For profile-based models, we transformed the predictions to a scalar through a global average along the length dimension.

CALCULATING SIMILARITY OF ATTRIBUTION SCORES TO ENSEMBLE AVERAGE

For each individual model, we calculated the root mean squared error (RMSE) of the attribution maps for the 1000 sequences evaluated between the individual models and the average attribution

maps of the teacher ensemble. Individual models refer to those trained with either DEGU distillation or standard training from random initializations. The teacher ensemble was calculated by averaging the attribution maps across the 10 individual models with standard training.

CALCULATING VARIABILITY OF ATTRIBUTION SCORES ACROSS DIFFERENT INITIALIZATIONS

For each of the 1000 sequences evaluated, we calculated the variance of their attribution scores for each nucleotide and position in each sequence across 10 individual models (trained with different random initializations). These per-nucleotide and per-position variances are then summed across the sequence to calculate the total variance, followed by a square root operation to provide a measure of the standard deviation of attribution scores across different initializations.

CONTROL EXPERIMENT WITH NORMALIZED ATTRIBUTION SCORES

For the control experiment that isolated mechanistic variability, we obtained the per-sequence maximum attribution score magnitude across nucleotides and positions for each of the 1000 sequences evaluated and then divided all attribution scores by this value to obtain an attribution-magnitude normalized set of attribution score.

EVALUATION OF THE SIZE OF THE TEACHER ENSEMBLE FOR DEEPSTARR

We trained an additional 15 DeepSTARR models using different random initializations using the entire STARR-seq training set for a total of 25 models. We then performed ensemble distribution distillation for subsets of 2, 3, 4, 5, 15, and 20 of these replicates, as well as for the entire set of 25 replicates. We evaluated the predictive accuracy of the activity predictions for the individual models in these ensembles, the ensemble average, and the distilled models derived from the respective teacher ensembles and compared them across different teacher ensemble sizes.

UNCERTAINTY-AWARE MODELS

HETEROSCEDASTIC REGRESSION

ResidualBind models trained with heteroscedastic regression utilized a Gaussian negative log-likelihood loss function. The final output layer was modified to a linear layer with 2 output heads representing the mean (μ) and log variance ($\log \sigma^2$). The use of log variance ensures numerical stability during training and guarantees positive variance predictions. The loss function is defined as:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \left[\log(2\pi\sigma_i^2) + \frac{(y_i - \mu_i)^2}{\sigma_i^2} \right] ,$$

where μ_i and $\log \sigma_i^2$ are predicted by the model and N is the number of samples in the batch. The model was trained using mini-batch stochastic gradient descent with the same optimizer, learning rate decay, and early stopping settings as used for standard training of the models described above. The variance predicted by heteroscedastic regression represents aleatoric uncertainty.

DEEP EVIDENTIAL REGRESSION

ResidualBind models trained with deep evidential regressionAmini et al. (2020) were modified so that their output layer considers the mean (μ) and log-variance ($\log \sigma^2$), which represents an estimate of the aleatoric uncertainty. The loss function is defined as:

$$\mathcal{L} = \mathbb{E} \left[\frac{1}{2} \left(\frac{(\mu - y)^2}{\sigma^2} + \log(2\pi) + \log \sigma^2 \right) \right] ,$$

where $\mathbb{E}[\cdot]$ denotes the expectation (average) over all dimensions and samples in the batch. The terms μ and $\log \sigma^2$ are predicted by the model and y is the true value.

MC DROPOUT

We implemented Monte Carlo (MC) dropout as described by Gal and Ghahramani Gal & Ghahramani (2016). This method leverages dropout at inference time to estimate predictive uncertainty.

For each input, we performed 100 stochastic forward passes through the model, with dropout remaining active during inference. We then calculated the mean and standard deviation across the predictions for each input. The mean prediction represents the model's best estimate, while the standard deviation quantifies the epistemic uncertainty associated with that prediction.

UNCERTAINTY CALIBRATION ANALYSIS

INTERVAL COVERAGE ANALYSIS

For each uncertainty quantification method, we calculated a 95% confidence interval using the model's prediction of sequence activity and uncertainty for the test sequences in model's corresponding dataset. For uncertainty quantification methods that yielded both aleatoric and epistemic uncertainty estimates, we calculated intervals based on each of the two different uncertainty estimates as well as the total uncertainty calculated as the sum of the variances, according to:

$$\sigma_T = \sqrt{(\sigma_E^2 + \sigma_A^2)}\,,$$

where σ_T is total uncertainty, σ_E^2 is epistemic uncertainty, and σ_A^2 is aleatoric uncertainty. For models where the uncertainty prediction was given as log-variance (i.e. models trained with heteroscedastic regression), the output was accordingly transformed for compatibility with total uncertainty as a measure of standard deviation. The interval coverage probability was calculated as the fraction of cases where the experimental activity value fell within the 95% confidence interval constructed from the predicted activity and uncertainty values. Assuming a Gaussian distribution, the 95% confidence interval was calculated as $\hat{\mu} \pm 1.96\hat{\sigma}$, where $\hat{\mu}$ and $\hat{\sigma}$ represent the estimates of activity and uncertainty for the method being evaluated.

CONFORMAL PREDICTIONS

Conformal prediction was used to calibrate the predicted uncertainties, according to:

$$\lambda = \text{quantile}_{1-\alpha}\left(\frac{|y_i - \hat{y}_i|}{\sigma_i}\right)\,,$$

where y_i are the true target values for the calibration sequence i, $\hat{y}_i$ are the predicted values for the calibration sequence i, $\hat{\sigma}_i$ are the uncertainty estimates for the calibration sequence i, and α is the desired confidence threshold, set to 0.05.

Calibration sequences were taken from the validation set. The calibration factor λ is then multiplied by the predicted uncertainty estimates for the test sequences.