References

[1] Ahmed Alaa and Mihaela Van Der Schaar. Discriminative Jackknife: Quantifying Uncertainty in Deep Learning via Higher-Order Influence Functions. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 165–174. PMLR, 2020.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.

[3] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. *CoRR*, abs/2009.14193, 2020.

[4] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

[5] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *arXiv*, abs/1903.04684, 2020.

[6] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence Functions in Deep Learning Are Fragile, 2021.

[7] M. J. Bayarri and J. O. Berger. The Interplay of Bayesian and Frequentist Analysis. *Statistical Science*, 19(1):58 – 80, 2004.

[8] Anthony Bellotti. Constructing normalized nonconformity measures based on maximizing predictive efficiency. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Giovanni Cherubin, editors, *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128 of *Proceedings of Machine Learning Research*, pages 41–54. PMLR, 09–11 Sep 2020.

[9] The boston housing dataset. `http://lib.stat.cmu.edu/datasets/boston`. Accessed: 2021-05-27.

[10] Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15, 2013.

[11] Adam Fisch, Tal Schuster, Tommi S. Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations*, 2021.

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *33rd International Conference on Machine Learning, ICML 2016*, 2016.

[13] Leying Guan. Conformal prediction with localization. *arXiv*, abs/1908.08558, 2020.

[14] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Science & Business Media, 2006.

[15] José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1861–1869. JMLR.org, 2015.

[16] Kin family of datasets. `http://www.cs.toronto.edu/~delve/data/kin/desc.html`. Accessed: 2021-05-27.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017.

[19] Benjamin Kompa, Jasper Snoek, and Andrew Beam. Empirical frequentist coverage of deep learning uncertainty quantification procedures. *CoRR*, abs/2010.03039, 2020.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

[21] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 2018.

[22] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):71–96, 2014.

[23] Zhen Lin, Cao Xiao, Lucas Glass, M. Brandon Westover, and Jimeng Sun. SCRIB: set-classifier with class-specific risk bounds for blackbox models. *CoRR*, abs/2103.03945, 2021.

[24] Elizbar Nadaraya. *Nonparametric Estimation of Probability Densities and Regression Curves*. Kluwer Academic Publishers, 1989.

[25] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Machine Learning: ECML 2002*, pages 345–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[27] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

[28] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O. Anatole von Lilienfeld. Electronic spectra from tddft and machine learning in chemical space. *The Journal of Chemical Physics*, 143(8):084111, 2015.

[29] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[30] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. PMID: 23088335.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[32] Sul and Elena Chow. Globalchem: A content variable store for chemistry! `https://github.com/Sulstice/global-chem`, 2021.

[33] Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candes, and Aaditya Ramdas. Conformal prediction under covariate shift, 2020.

[34] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.

[35] Bike sharing data set. `https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset`. Accessed: 2021-05-27.

[36] Concrete compressive strength data set. `http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength`. Accessed: 2021-05-27.

[37] Energy efficiency data set. `https://archive.ics.uci.edu/ml/datasets/energy+efficiency`. Accessed: 2021-05-27.

[38] Yacht hydrodynamics data set. `http://archive.ics.uci.edu/ml/datasets/yacht+hydrodynamics`. Accessed: 2021-05-27.

[39] Vladimir Vovk. Conditional validity of inductive conformal predictors. In Steven C. H. Hoi and Wray Buntine, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pages 475–490, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR.

[40] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer US, 2005.

[41] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964.

[42] Kilian Q. Weinberger and Gerald Tesauro. Metric learning for kernel regression. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 612–619, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[43] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

[44] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[45] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.

[46] I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998.

A Proofs

A.1 Proof for Theorem 3.1

In this section, we will prove Theorem 3.1. The key idea behind the proof is that since $\hat{\mu}$ (let it be $\hat{\mu}^{NN}$ or $\hat{\mu}^{KR}$) and $K_{\mathbf{f}}$ are independent of $\mathcal{S}_{\text{conformal}}$, the residuals (R_i) collected on $\mathcal{S}_{\text{conformal}}$ follow the same distribution as a new test residual. Thus, re-weighting $\mathcal{P}_X$ by $K_{\mathbf{f}}$ precisely fits into the covariate-shift setting studied by [33] (further studied in [13]), which in turn implies that under the new (localized) distribution for X, the coverage guarantee holds (Theorem 3.1).

We first introduce a few definitions following the same notation as in [13]. To begin, we define the score function as $V(x, y) := |y - \hat{\mu}(x)|$. Then, we write the localizer function as $H(x, x') := K_{\mathbf{f}}(x, x')$. For convenience, we also will rewrite subscripts of the data, so we have $Z'_1 = (X'_1, Y'_1), \ldots, Z'_{m+1} = (X'_{m+1}, Y'_{m+1})$, where Z'_i is just Z_{n+i} for $i \in [m]$ and Z'_{m+1} is Z_{N+1}. For $\{Z'_i\}_{i=1}^{m+1}$, both V and H would be considered *fixed* because the training did not use any information from $\mathcal{S}_{\text{conformal}}$. [13] allows for a more general form of H that can depend on $\mathcal{S}_{\text{conformal}}$, which is not needed in our setting.

We proceed to define the weighted residual distributions like in [13]:

$$\hat{\mathcal{F}}_i := \sum_{j=1}^{m+1} p^H_{i,j} \delta_{V(X'_j, Y'_j)} \tag{17}$$

$$\text{where } p^H_{i,j} := \frac{H(X'_i, X'_j)}{\sum_{k=1}^{m+1} H(X'_i, X'_k)} \tag{18}$$

Finally, $\hat{\mathcal{F}}$ is defined as $p^H_{m+1,m+1}\delta_\infty + \sum_{i=1}^{m} p^H_{m+1,i}\delta_{V(X'_i, Y'_i)}$. $V(X'_{m+1}, Y'_{m+1})$ can be considered set to ∞, because we don't know the value of Y'_{m+1} and want to be conservative.

Now, our construction of the PI could be rewritten in the following form:

$$\hat{C}^{LVD}_\alpha(X'_{m+1}) := \{y : V(X'_{m+1}, y) \leq Q(1 - \alpha, \hat{\mathcal{F}})\} \tag{19}$$

This is precisely the setup of Theorem 5.1 in [13], and Theorem 3.1 follows from Theorem 5.1 in [13].

A.2 Asymptotic Conditional Validity (Theorem 3.2)

Before we discuss the asymptotic property of $\hat{C}^{LVD}$, we formally define asymptotic conditional validity (from [22]).

Definition 1. *(Asymptotic Conditional Validity) Given training data* $(X_1, Y_1), \ldots, (X_m, Y_m)$*, a PI estimator* $\hat{C}_{m,\alpha}$ *is asymptotically conditionally valid if*

$$\sup_x \Big[\mathbb{P}\{Y_{m+1} \notin C_{m,\alpha}(x) | X_{m+1} = x\} - \alpha\Big]_+ \xrightarrow{\mathcal{P}} 0 \tag{20}$$

as $m \to \infty$*, where the sup is taken over the support of* $\mathcal{P}_X$.

Here, we add the subscript m to $\hat{C}_\alpha$ to emphasize the dependence on the sample size. If a PI estimator is asymptotically conditionally valid at level $1 - \alpha$, then given enough samples (as $m \to \infty$), the probability of $\hat{C}_{m,\alpha}$ missing the next response Y_{m+1} converges to α in probability. Note that LVD has an implicit assumption that the embedding function $\mathbf{f}$ (after some transformation) maps similar data close together. But with Theorem 3.2 such an assumption is not critical as the size of the dataset increases.

To facilitate the discussion, we add a subscript m and denote the PI given by LVD as $\hat{C}^{LVD}_{m,\alpha}$. With the setup mentioned in Section A.1, we obtain a result similar to Theorem 5.1 (b) in [13] to for $\hat{C}^{LVD}_{m,\alpha}$ as well.

Assumptions: We need to make the following assumptions:

(1) Denote $W := \mathbf{f}(X)$ as a new random variable in $\mathbb{R}^h$. W is (assumed to be) on $[0,1]^h$ with marginal density bounded from two sides by two constants $b_1 < b_2$. In other words, $0 < b_1 \leq p_W(w) \leq b_2 < \infty$.
(2) The conditional density of R (the residual) given W is Lipschitz in W. In other words, $\forall w, w'$, $\|p_{R|W}(\cdot|w) - p_{R|W}(\cdot|w')\|_\infty \leq L\|w - w'\|$.

As might be clear, (1) and (2) are standard regularity assumptions (as in [13, 22]), but stated for our setting. For assumption (1), if W does not fall in $[0,1]^h$, we can easily fix it by adding a normalization layer to $\mathbf{f}$. Compared with [13, 22], (2) is not any less likely to hold, as we usually only have one linear layer after $\mathbf{f}$ in $\hat{\mu}^{NN}$.

Bandwidth (h): To clearly state the theorem, we also need to decompose/unfold our transform matrix $\mathbf{A}$ into two steps - projection and rescaling: $\mathbf{A}(w - w') := \frac{1}{h}\mathbf{A_1}(w - w')$, where $\|\mathbf{A}_1\|_2 = 1$. Note that in our learning, we are mostly learning $\mathbf{A}_1$, and h is in fact *chosen*. In our experiment, we implicitly folded h into $\mathbf{A}$, as changing h entails making an explicit decision on how "local" one wants the coverage to be when the data is limited, and we do not have a strong prior on this. However, for the sake of this discussion, as $N \to \infty$, if we keep the same ratio between $n = |\mathcal{S}_{\text{embed}}|$ and $m = |\mathcal{S}_{\text{conformal}}|$, then:

- $\mathbf{A}_1$ would converge to some fixed unit-norm matrix in $\mathbb{R}^{h \times k}$, and
- we could let $h \to 0$ like in [13] and [22], because if we have $m \to \infty$, then the number of validation residuals is large, so we could afford a much more "local" validity with few infinitely wide PIs.

With the assumptions stated above, we are in a position to state the following theorem regarding the asymptotic conditional validity of $\hat{C}^{LVD}_{m,\alpha}$:

Theorem A.1. *(Asymptotic Conditional Validity. Re-statement of Theorem 3.2): With assumptions (1) and (2), and $m \to \infty$, if we also let $h \to 0$, then*

$$\left[\alpha - \mathbb{P}\{Y'_{m+1} \in \hat{C}^{LVD}_{m,\alpha}(X'_{m+1})\}\right]_+ \xrightarrow{\mathcal{P}} 0. \tag{21}$$

The proof is essentially the same as that in [13], with the key difference that in [13], the Gaussian kernel only has one bandwidth h, which goes to 0 asymptotically. This has been discussed in the "Bandwidth (h)" section above. Note the key difference between Theorem A.1 and 3.1 is that the response Y'_{m+1} now belongs to X'_{m+1}, which is used to construct the PI.

B Additional Experimental Details

B.1 Training Details

As noted in the paper, the DNN used for most datasets (except QM8 and QM9) has 2 layers, 100 hidden nodes, and uses ReLU for the activation function. This is the same architecture as in [1], but with the difference that the activation is ReLU instead of tanh. We make this choice because the code accompanying [1] uses ReLU, and tanh does not train for most of the datasets in our experiments. Recall that the learnable matrix $\mathbf{A}$ reduces dimension from h to k. For QM8 and QM9, please refer to [45] for a detailed description of the architecture and training protocols. We make the following modifications in order to run some baselines:

- MADSplit: We train a second model after the model in [45] that has the same architecture and training protocol, but tries to predict the absolute error of the first model.
- CQR: We replace the MSE loss with the "pinball" loss mentioned in [29] and simultaneously train two quantiles for the same α. For different α, we re-train a model.
- DE: We replace the loss with the negative log-likelihood (NLL) loss as suggested in [20], and train an ensemble of 5 models for each experiment.

For all experiments, h is given by the DNN, and we set $k = 10$. The training of the kernel follows the following protocol: we first take embedding from the training data, compute and fix the mean μ_i and standard deviation s_i for each dimension $i \in [h]$. Dimensions with standard deviation $<$1e-3 are ignored as they are most likely dead nodes (due to ReLU). The embeddings are then always normalized using μ_i and s_i before passing through $\mathbf{A}$.

$\mathbf{A}$ is implemented as a `torch.nn.Linear` layer using PyTorch[26] and follows the default initialization. We restrict the kernel regression to use the top 3000 (or all) similar data points so the computation can be fast (like in [42]). We use an Adam optimizer [17] implemented in PyTorch, with a learning rate set to 1e-2, and batch size 100. We repeat the process for 1000 up to batches, and stop early if the loss does not improve for 50 consecutive batches.

For each setup, we repeat the experiment 10 times by randomly re-splitting training, validation, and test set with random seed from 0 to 9. For LVD, MADSplit, and CQR (which require a hold-out set for conformal prediction), we use 60% for training, 20% for validation/hold-out set, and 20% for test. For all other methods, we use 80% for training and 20% for testing.

B.2 Average PI Width

It is hard to compare efficiency because LVD achieves a much more demanding type of coverage, MADSplit and CQR achieve marginal coverage, and the rest of the methods are not valid (thus not comparable). We thus restrict the comparison to only valid methods (LVD, MADSplit, and CQR) and the subset of data for which all PIs are finite in Table 5. We can see that, as expected, LVD tends to give infinite PI for small datasets at 90% target level ("# finite" is low for a few datasets), because it requires some weighted observation in a local neighborhood. (Note that the # of finite PIs could be tuned by a bandwidth h as discussed in Section A.2.) However, despite providing a stronger coverage guarantee, LVD still managed to be the most efficient on Bike and QM9.

The most efficient method seems to be CQR, but the results are not very stable (very wide PIs for CQR in the Bike dataset, for example), and most of the time the difference in average width is not significant. However, as noted earlier in the main text, the potential efficiency of CQR comes with a huge cost: CQR requires re-training the model for each α. Moreover, there is no guarantee that the estimate of the lower bound of the PI is actually lower than the upper bound ("quantile crossing", see [29]), nor that a mean estimate actually falls in the PI either. In our experiments, we had to take the mean of the lower and upper bound as the mean estimator to ensure the mean estimator is always within the PI.

We also include the average width of all baselines in Table 6 for reference, although it is not very meaningful to compare valid and non-valid methods.

Table 5: Average width of different conformal methods. Width significantly shorter than the second-best at $p = 0.05$ are in bold.

	50%-PI Width				90%-PI Width			
Data (Count)	# finite	LVD	MADSplit	CQR	# finite	LVD	MADSplit	CQR
Yacht(62)	61.90±0.32	3.99±0.79	3.17±0.84	2.82±0.74	40.90±2.85	3.47±1.36	3.29±1.04	4.52±2.08
Housing(101)	98.30±3.06	6.70±0.97	6.02±1.23	**4.98**±0.72	69.00±19.11	15.94±2.62	16.81±7.44	**13.70**±1.84
Energy(154)	154.00±0.00	6.06±1.41	5.77±1.37	5.18±1.47	145.10±11.05	12.89±2.02	12.19±2.71	13.76±2.80
Bike(3476)	3475.20±1.23	0.06±0.05	0.07±0.05	5.62±4.41	3467.50±4.40	0.15±0.13	0.19±0.11	33.65±21.52
Kin8nm(1638)	1610.10±10.18	0.14±0.01	0.12±0.01	0.12±0.01	938.00±123.72	0.34±0.02	0.28±0.02	0.28±0.02
Concrete(206)	200.10±4.01	10.92±1.85	9.77±1.66	9.35±2.84	133.80±20.13	27.93±3.59	21.79±2.93	22.87±3.48
QM8*(4357)	4317.33±22.90	0.02±0.01	0.02±0.01	0.04±0.01	4041.63±136.85	0.05±0.03	0.05±0.03	0.11±0.03
QM9*(26744)	26616.72±39.77	5.12±13.17	5.75±14.78	37.32±65.01	26146.95±151.58	15.06±38.94	14.77±37.04	129.63±207.46

Table 6: Average width of all baselines methods, without restriction to the subsample for which LVD gives finite PIs.

Width @ 50%	MADSplit	CQR	DJ	DE	MCDP	PBP
Yacht	3.18±0.82	2.83±0.72	19.10±1.26	5.26±0.78	14.34±0.71	2.16±0.31
Housing	6.06±1.22	5.00±0.71	11.50±1.41	10.23±1.50	30.58±0.33	0.74±0.08
Energy	5.77±1.37	5.18±1.47	9.83±1.39	10.52±1.59	30.14±0.24	0.78±0.04
Bike	0.07±0.05	5.62±4.41	0.14±0.07	13.62±6.61	115.47±0.84	0.84±0.27
Kin8nm	0.12±0.01	0.12±0.01	0.25±0.02	0.80±0.03	0.98±0.02	1.29±0.14
Concrete	9.84±1.70	9.39±2.82	47.98±84.01	18.33±2.96	47.82±0.30	0.78±0.06
QM8*	0.05±0.03	0.11±0.03	–	42.17±28.01	–	–
QM9*	14.77±37.04	129.63±207.46	–	465.17±919.56	–	–

Width @ 90%	MADSplit	CQR	DJ	DE	MCDP	PBP
Yacht	8.02±0.98	12.31±1.79	73.14±1.75	13.13±1.24	34.96±1.73	5.26±0.77
Housing	18.71±9.91	15.10±1.79	26.31±1.87	24.97±2.58	74.57±0.81	1.82±0.19
Energy	12.24±2.78	13.75±2.88	18.54±2.10	25.65±5.58	73.50±0.60	1.91±0.09
Bike	0.19±0.11	33.96±21.87	0.32±0.18	38.50±13.26	281.24±1.66	2.04±0.65
Kin8nm	0.31±0.02	0.32±0.01	0.48±0.03	1.89±0.15	2.39±0.06	3.15±0.35
Concrete	22.52±2.93	23.29±3.32	199.19±369.33	44.02±4.73	116.62±0.74	1.90±0.16
QM8*	0.05±0.03	0.11±0.03	–	42.17±28.01	–	–
QM9*	14.77±37.04	129.63±207.46	–	465.17±919.56	–	–

B.3 Additional Results of Different Variants of LVD

Although we consider MADSplit as a baseline, our method could be combined with it as well, by simply replacing R_i with a normalized $R'_i := \frac{y_{n+i} - \hat{y}_{n+i}}{\hat{\sigma}(x_{n+i})}$ like that in MADSplit. One key observation is that using embedding given by a pre-trained DL model can simultaneously keep most of the performance of the base model and combine it with many conformal methods with acceptable overhead.

In this section, we will change different settings of LVD and compare the effects. Specifically, there are 3 independent choices:

- Whether we use the kernel regression prediction $\hat{y}^{KR}$ or the base DNN predictor $\hat{\mu}^{NN}$ (KR vs. NN)
- Whether we apply the smoothness requirement as mentioned in Section 3.3 (No-smooth vs. Smooth)
- Whether we normalize the residuals by an extra prediction of MAD or not. We will denote the version described in the main text as "base". For the MAD-Normalized case ("MN"), similar to MADSplit [21, 8], the non-conformity score, and the final PI construction, are replaced by

$$R'_i := \frac{y_{n+i} - \hat{y}_{n+i}}{\hat{\sigma}(x_{n+i})} \tag{22}$$

$$\hat{C}^{MN}_{\alpha}(X_{N+1} := \left\{ y \in \mathbb{R} : |y - \hat{y}_{N+1}| \leq \frac{1}{\hat{\sigma}(X_{N+1})} Q\left(1 - \alpha, w_{N+1}\delta_{\infty} + \sum_{i=1}^{m} w_{n+i}\delta_{R'_i}\right)\right\} \tag{23}$$

This potentially can make the PI more discriminative by modeling the heteroscedasticity explicitly.

As a reminder, all results shown in the main text are using $\hat{\mu}^{NN}$, with smoothing, and not normalized by MAD prediction (NN, Smooth, NM). Also, all choices will not break any theoretical guarantees, including Theorem 3.1 and 3.2.

The results are presented in Table 7 and 8, with the version shown in the main text boxed. All methods achieve target coverage rates as measured by MCR and TCR empirically. In general, we found that using $\hat{y}^{KR}$ tends to give higher AUROC, with similar or lower MAD. It should be noted that the MAD prediction in "MN" requires a base classifier, which is $\hat{y}^{NN}$ in our case. In other words, there is a mismatch in the "MN" version with $\hat{y}^{KR}$. We conjecture that if the MAD predictor is properly trained for $\hat{y}^{KR}$, the AUROC for this combination would be even higher (at no cost to other metrics).

We also include the average width and count of finite PIs in Table 9 and 10. For most experiments adding smoothness requirement and using $\hat{y}^{KR}$ seems to achieve narrow PI, high AUROC, and low MAD. As noted earlier, training a separate model to model the residual of $\hat{y}^{KR}$ might give additional discrimination (and possibly narrower PIs as well).

Table 7: MCR and TCR for different variants of LVD.

	$\hat{y}^{KR}$				$\hat{y}^{NN}$			
	No-smooth		Smooth		No-smooth		Smooth	
MCR	MN	base	MN	base	MN	base	MN	$\boxed{base}$
Yacht	**96.6**±4.5	**97.4**±2.0	**95.2**±4.6	**95.5**±2.3	**96.1**±4.9	**97.9**±1.7	**94.7**±5.0	**96.8**±2.2
Housing	**96.8**±3.4	**97.1**±2.8	**96.0**±3.9	**95.7**±3.5	**97.3**±2.2	**97.8**±2.1	**96.1**±2.6	**96.8**±2.9
Energy	**92.8**±2.7	**92.4**±2.9	**92.5**±2.6	**92.4**±2.9	**94.0**±1.7	**94.1**±1.6	**93.9**±1.7	**94.0**±1.6
Bike	**91.6**±1.2	**93.8**±0.8	**91.6**±1.2	**94.1**±0.8	**90.6**±0.5	**90.5**±0.8	**90.4**±0.6	**90.4**±0.8
Kin8nm	**100.0**±0.0	**100.0**±0.0	**97.9**±0.7	**97.9**±0.8	**100.0**±0.0	**100.0**±0.0	**97.9**±0.6	**98.0**±0.6
Concrete	**99.6**±0.7	**99.6**±0.8	**96.7**±2.3	**97.0**±1.1	**99.7**±0.6	**99.6**±0.7	**97.0**±2.1	**97.4**±1.3
QM8*	**94.9**±1.4	**95.3**±1.3	**92.3**±0.8	**92.9**±0.9	**94.7**±1.5	**95.1**±1.4	**92.0**±0.9	**92.6**±0.9
QM9*	**94.0**±1.7	**94.1**±1.6	**90.6**±0.4	**90.4**±0.5	**93.5**±1.5	**93.7**±1.5	**90.3**±0.4	**90.3**±0.6
TCR								
Yacht	**96.9**±4.0	**96.9**±5.4	**93.1**±7.6	**94.6**±5.2	**95.4**±7.4	**99.2**±2.4	**93.8**±7.1	**98.5**±3.2
Housing	**95.7**±4.2	**96.2**±4.4	**93.3**±9.3	**91.4**±8.3	**98.1**±3.3	**97.1**±4.0	**98.1**±2.5	**96.2**±4.4
Energy	**86.8**±6.2	83.5±9.7	85.8±6.1	83.2±10.2	**88.1**±4.8	**87.1**±5.9	**87.7**±4.8	**86.8**±5.8
Bike	**91.8**±1.0	**91.9**±2.1	**90.9**±1.6	**91.6**±2.7	**92.0**±1.3	**90.8**±1.5	**91.6**±1.3	**90.2**±1.7
Kin8nm	**100.0**±0.0	**100.0**±0.0	**95.7**±1.6	**95.0**±2.1	**100.0**±0.0	**100.0**±0.0	**97.1**±1.5	**97.2**±1.6
Concrete	**99.0**±2.1	**99.3**±1.6	**93.9**±4.5	**94.4**±3.6	**99.5**±1.0	**99.5**±1.5	**96.8**±3.8	**97.1**±3.4
QM8*	**94.6**±2.1	**95.6**±2.4	**90.4**±2.0	**91.4**±2.6	**94.9**±1.9	**94.8**±2.2	**91.2**±1.7	**90.8**±1.9
QM9*	**94.4**±4.4	**94.3**±4.5	**88.5**±3.7	**88.0**±3.7	**94.9**±3.1	**94.8**±3.4	**90.1**±2.3	**89.7**±2.5

Table 8: AUROC and MAD for different variants of LVD. Best AUROCs are in bold, and all are significantly higher than 50 (at $p = 0.05$). For MAD, the best for each task, if significantly better than the second-best (at $p = 0.05$), are in bold.

	$\hat{y}^{KR}$				$\hat{y}^{NN}$			
	No-smooth		Smooth		No-smooth		Smooth	
AUROC	MN	base	MN	base	MN	base	MN	$\boxed{base}$
Yacht	71.1±7.1	74.2±3.5	61.5±12.4	67.5±7.1	81.0±6.1	**83.8**±5.4	80.9±6.1	83.5±5.8
Housing	58.7±7.1	62.0±6.6	61.4±6.7	**64.4**±5.6	62.6±7.9	60.0±7.0	62.1±9.0	59.2±8.5
Energy	61.8±5.0	60.8±2.9	63.3±5.3	62.9±4.6	74.3±7.2	73.5±6.3	**74.3**±7.2	73.5±6.3
Bike	73.5±7.8	86.6±3.6	73.7±7.5	**87.5**±3.2	72.3±8.5	68.1±11.1	72.4±8.5	68.2±11.0
Kin8nm	55.7±1.8	55.9±2.1	60.5±1.9	**61.8**±1.6	57.1±2.4	56.8±2.4	61.6±1.6	60.3±1.1
Concrete	60.4±3.5	60.1±3.4	62.8±4.6	61.8±5.0	62.7±8.4	62.4±8.4	**65.4**±6.1	64.0±6.1
QM8*	73.2±9.6	72.8±11.9	**75.9**±9.0	75.2±11.9	72.9±7.7	71.3±9.5	74.1±6.9	71.3±9.4
QM9*	**68.2**±7.6	67.3±8.9	66.5±3.5	66.4±5.4	66.2±3.5	64.1±3.7	66.3±3.5	62.7±3.6
MAD								
Yacht	**0.79**±0.09	**0.79**±0.09	1.14±0.13	1.14±0.13	1.90±0.48	1.90±0.48	1.90±0.48	1.90±0.48
Housing	2.86±0.31	2.86±0.31	3.00±0.32	3.00±0.32	3.31±0.53	3.31±0.53	3.31±0.53	3.31±0.53
Energy	2.34±0.07	2.34±0.07	2.35±0.08	2.35±0.08	2.99±0.75	2.99±0.75	2.99±0.75	2.99±0.75
Bike	2.47±0.72	2.47±0.72	3.79±0.49	3.79±0.49	**0.04**±0.03	**0.04**±0.03	**0.04**±0.03	**0.04**±0.03
Kin8nm	**0.06**±0.00	**0.06**±0.00	0.07±0.00	0.07±0.00	0.07±0.00	0.07±0.00	0.07±0.00	0.07±0.00
Concrete	**4.76**±0.26	**4.76**±0.26	5.20±0.29	5.20±0.29	5.44±0.53	5.44±0.53	5.44±0.53	5.44±0.53
QM8*	0.01±0.01	0.01±0.01	0.01±0.01	0.01±0.01	0.01±0.01	0.01±0.01	0.01±0.01	0.01±0.01
QM9*	3.58±9.71	3.58±9.71	4.92±11.39	4.92±11.39	3.69±9.09	3.69±9.09	3.69±9.09	3.69±9.09

Table 9: Counts of finite PIs and average width for different variants of LVD (restricted to the subset for which all PIs are finite), with $\alpha = 0.5$. In the count table, the size of the test set is included in the parenthesis, and the lowest count (which is used for width computation) is underscored.

	$\hat{y}^{KR}$				$\hat{y}^{NN}$			
	No-smooth		Smooth		No-smooth		Smooth	
# finite @ 50%	MN	base	MN	base	MN	base	MN	$\boxed{base}$
Yacht(62)	60.7±1.9	60.7±1.9	61.9±0.3	61.9±0.3	60.7±1.9	60.7±1.9	61.9±0.3	61.9±0.3
Housing(101)	93.5±5.7	93.5±5.7	98.3±3.1	98.3±3.1	93.5±5.7	93.5±5.7	98.3±3.1	98.3±3.1
Energy(154)	154.0±0.0	154.0±0.0	154.0±0.0	154.0±0.0	154.0±0.0	154.0±0.0	154.0±0.0	154.0±0.0
Bike(3476)	3473.0±2.8	3473.0±2.8	3475.2±1.2	3475.2±1.2	3473.0±2.8	3473.0±2.8	3475.2±1.2	3475.2±1.2
Kin8nm(1638)	844.3±181.0	844.3±181.0	1610.1±10.2	1610.1±10.2	844.3±181.0	844.3±181.0	1610.1±10.2	1610.1±10.2
Concrete(206)	176.5±21.7	176.5±21.7	200.1±4.0	200.1±4.0	176.5±21.7	176.5±21.7	200.1±4.0	200.1±4.0
QM8*(4357)	4002.4±210.2	4002.4±210.2	4317.3±22.9	4317.3±22.9	4002.4±210.2	4002.4±210.2	4317.3±22.9	4317.3±22.9
QM9*(26744)	25376.7±792.3	25376.7±792.3	26616.7±39.8	26616.7±39.8	25376.7±792.3	25376.7±792.3	26616.7±39.8	26616.7±39.8
Width @ 50%								
Yacht	1.8±0.5	**1.7**±0.5	2.2±0.5	2.1±0.4	4.5±0.9	4.4±0.9	4.1±0.9	3.8±0.9
Housing	5.9±0.6	5.7±0.8	5.5±0.6	**5.3**±0.7	7.2±1.1	7.0±1.1	6.7±1.3	6.4±1.0
Energy	4.3±0.5	**4.3**±0.3	4.3±0.4	4.3±0.3	6.2±1.5	6.1±1.4	6.2±1.5	6.1±1.4
Bike	5.4±1.5	4.8±1.3	8.3±1.4	7.3±0.9	0.1±0.1	0.1±0.0	0.1±0.1	**0.1**±0.0
Kin8nm	0.2±0.0	0.2±0.0	0.1±0.0	**0.1**±0.0	0.2±0.0	0.2±0.0	0.1±0.0	0.1±0.0
Concrete	11.5±1.2	11.2±1.1	10.1±1.1	**9.8**±1.1	13.1±2.1	12.8±2.2	10.8±1.8	10.5±1.8
QM8*	0.0±0.0	0.0±0.0	0.0±0.0	**0.0**±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
QM9*	6.1±17.3	5.5±15.6	7.6±18.6	6.8±16.7	6.3±15.8	5.6±14.2	5.7±14.7	**5.0**±12.9

Table 10: Same as Table 9, but with $\alpha = 0.1$.

	$\hat{y}^{KR}$				$\hat{y}^{NN}$			
	No-smooth		Smooth		No-smooth		Smooth	
# finite @ 90%	MN	base	MN	base	MN	base	MN	$\boxed{base}$
Yacht(62)	35.1±5.8	35.1±5.8	40.9±2.8	40.9±2.8	35.1±5.8	35.1±5.8	40.9±2.8	40.9±2.8
Housing(101)	54.5±22.1	54.5±22.1	69.0±19.1	69.0±19.1	54.5±22.1	54.5±22.1	69.0±19.1	69.0±19.1
Energy(154)	145.1±11.0	145.1±11.0	145.1±11.0	145.1±11.0	145.1±11.0	145.1±11.0	145.1±11.0	145.1±11.0
Bike(3476)	3458.5±11.4	3458.5±11.4	3467.5±4.4	3467.5±4.4	3458.5±11.4	3458.5±11.4	3467.5±4.4	3467.5±4.4
Kin8nm(1638)	0.4±1.0	0.4±1.0	938.0±123.7	938.0±123.7	0.4±1.0	0.4±1.0	938.0±123.7	938.0±123.7
Concrete(206)	28.8±24.8	28.8±24.8	133.8±20.1	133.8±20.1	28.8±24.8	28.8±24.8	133.8±20.1	133.8±20.1
QM8*(4357)	2936.9±587.9	2936.9±587.9	4041.6±136.8	4041.6±136.8	2936.9±587.9	2936.9±587.9	4041.6±136.8	4041.6±136.8
QM9*(26744)	21347.5±2850.8	21347.5±2850.8	26147.0±151.6	26147.0±151.6	21347.5±2850.8	21347.5±2850.8	26147.0±151.6	26147.0±151.6
Width @ 90%								
Yacht	5.69±2.98	**2.21**±0.58	7.68±5.57	2.33±0.41	5.82±3.29	3.03±1.40	4.85±2.08	2.97±1.41
Housing	32.33±34.63	14.14±2.09	22.67±21.69	**13.29**±1.85	43.46±56.19	15.50±2.89	25.76±20.69	14.48±2.64
Energy	14.61±7.54	**12.47**±1.48	15.07±8.52	12.49±1.47	15.94±10.07	12.91±2.03	15.92±10.08	12.89±2.02
Bike	21.46±11.96	8.33±2.43	35.62±21.29	11.78±1.72	0.19±0.11	0.15±0.13	0.19±0.12	**0.15**±0.13
Kin8nm	0.44±0.03	0.36±0.02	0.27±0.12	0.25±0.03	0.39±0.08	0.37±0.02	0.25±0.11	**0.24**±0.02
Concrete	27.25±4.77	26.30±4.71	**20.55**±4.06	21.76±2.67	28.79±6.79	28.25±5.16	21.56±4.00	23.41±3.50
QM8*	0.05±0.03	0.04±0.02	0.04±0.02	**0.04**±0.02	0.05±0.03	0.04±0.02	0.04±0.02	0.04±0.02
QM9*	14.87±41.31	15.01±42.57	17.86±43.59	17.20±43.19	14.84±38.27	15.22±40.44	**13.65**±35.71	14.04±37.44

B.4 Additional Results on QM8/QM9 sub-tasks

Table 12 and 11 show the metrics for validity and discrimination, respectively, of different variants of LVD, and the two valid baselines. Table 13 shows the number of widths of PIs by different methods on the QM subtasks. Table 14 shows the coverage rates for a list of functional groups from the OPENSMILES project[8]. We keep only the subset of data whose original SMILES representation contains the corresponding functional group's SMILES representation, and compute the average coverage rate for each of the twelve targets of QM9 dataset[9]. If LVD is actually conditionally valid, then the conditional coverage rate should not be significantly lower than the target (90%). Again, it is worth noting that LVD is only *approximately* conditionally valid, and the raw SMILES functional groups were not used anywhere in the entire pipeline. However, LVD is still almost always valid empirically.

---

[8] `http://opensmiles.org/opensmiles.html`

[9] We only did this for QM9 because the size of QM8 is not enough for this task.

Table 11: AUROC and MAD for QM8 and QM9 sub-tasks. The best AUROCs are in bold, and AUROCs **not** significantly higher than 50 at $p = 0.05$ are underscored. For MAD, the best MADs, if significantly better than the second baseline at $p = 0.05$, are in bold.

	LVD								Conformal Baselines	
	$\hat{y}^{KR}$				$\hat{y}^{NN}$					
	No-smooth		Smooth		No-smooth		Smooth			
AUROC	MN	base	MN	base	MN	base	MN	$\boxed{base}$	MADSplit	CQR
QM8(E1-CC2)	64.3±1.4	61.8±1.5	67.9±0.9	64.9±1.0	66.2±1.0	63.2±1.2	**68.2**±0.9	62.9±1.4	67.7±0.8	57.9±3.8
QM8(E2-CC2)	62.5±1.5	60.2±1.3	66.7±0.8	63.2±1.2	64.3±1.2	61.1±1.2	**66.8**±1.1	61.7±1.2	66.2±1.0	56.8±3.9
QM8(f1-CC2)	83.8±2.1	85.7±2.5	85.7±1.0	**87.9**±0.8	80.2±1.0	80.5±1.2	80.5±1.2	80.1±1.4	80.0±1.0	70.5±2.9
QM8(f2-CC2)	81.4±2.6	82.9±2.2	84.3±1.3	**86.0**±0.7	81.7±1.0	82.1±0.7	82.1±0.8	82.3±0.5	80.9±1.1	78.5±5.7
QM8(E1-PBE0)	64.7±1.6	61.8±1.6	67.6±0.8	63.9±1.6	66.2±1.1	62.6±1.1	**68.3**±0.9	62.6±1.1	67.6±0.9	55.2±5.0
QM8(E2-PBE0)	63.9±1.4	60.2±1.2	66.0±1.0	61.8±1.2	65.1±1.3	61.3±1.4	**66.6**±1.2	61.2±1.5	66.3±1.2	55.9±2.2
QM8(f1-PBE0)	83.8±2.2	86.0±2.0	85.2±1.4	**87.8**±0.7	79.0±1.1	78.8±1.0	79.1±1.1	78.1±1.7	79.0±1.0	67.2±2.7
QM8(f2-PBE0)	80.1±2.3	82.1±2.8	82.8±1.1	**85.2**±0.9	80.8±0.9	81.0±0.8	81.2±1.2	81.1±1.1	80.3±1.0	79.9±3.1
QM8(E1-PBE0.1)	64.8±1.6	61.8±1.6	67.6±0.9	63.9±1.6	66.3±1.4	62.6±1.1	**68.2**±0.7	62.4±0.8	67.4±0.8	56.9±2.5
QM8(E2-PBE0.1)	63.8±1.4	60.2±1.2	65.8±1.0	61.8±1.2	64.6±1.3	60.8±1.5	**66.1**±0.9	60.8±1.5	66.0±1.1	55.2±2.3
QM8(f1-PBE0.1)	83.8±2.2	86.0±2.0	85.1±1.4	**87.8**±0.7	79.5±1.5	79.3±1.2	79.7±1.4	78.7±1.8	79.4±1.3	68.1±3.8
QM8(f2-PBE0.1)	80.1±2.3	82.1±2.8	82.9±1.1	**85.2**±0.9	81.1±0.7	81.3±0.8	81.6±1.0	81.4±1.1	80.6±0.9	76.7±6.9
QM8(E1-CAM)	63.4±1.6	61.4±1.2	67.7±1.1	64.8±1.0	65.4±1.2	62.2±1.0	**68.2**±1.0	63.2±0.7	67.4±1.0	58.0±3.9
QM8(E2-CAM)	63.9±2.2	61.2±1.7	66.6±0.8	63.3±1.8	65.1±2.1	61.7±1.6	**66.9**±1.4	62.4±2.1	66.5±1.5	56.8±3.3
QM8(f1-CAM)	85.8±1.5	87.7±1.5	87.2±1.0	**89.5**±0.5	78.7±1.2	79.2±1.2	78.7±1.3	78.8±1.2	78.4±1.2	73.1±2.0
QM8(f2-CAM)	81.5±2.1	83.0±2.2	84.8±1.0	**86.8**±0.8	82.7±1.1	82.9±1.0	83.2±1.0	83.3±1.1	82.1±1.0	81.4±2.3
QM9(mu)	71.7±0.6	67.6±0.5	71.8±1.0	66.6±1.7	72.6±1.1	68.2±0.5	**73.9**±0.7	68.0±1.1	73.7±0.7	57.4±3.4
QM9(alpha)	61.6±1.2	60.3±1.3	**66.2**±1.3	65.9±1.9	65.3±0.8	63.2±0.9	65.3±0.6	61.3±1.4	64.0±0.5	$\underline{45.9}$±7.7
QM9(homo)	61.2±0.9	58.3±0.5	61.9±0.5	58.4±0.8	62.2±0.4	58.6±0.5	**62.9**±0.3	58.1±0.9	62.6±0.4	$\underline{38.7}$±20.8
QM9(lumo)	60.6±0.7	58.6±0.7	61.5±0.7	59.3±0.7	61.2±0.8	58.5±0.4	**62.4**±0.4	58.1±0.4	62.2±0.4	$\underline{58.4}$±20.1
QM9(gap)	62.3±1.0	60.4±1.0	62.8±0.5	60.5±0.8	62.8±0.9	60.1±0.7	**63.5**±0.5	59.6±0.6	63.1±0.5	$\underline{60.2}$±25.0
QM9(r2)	67.8±1.1	64.1±1.1	68.5±1.2	65.0±1.0	69.3±0.7	64.7±0.6	**69.8**±0.5	63.1±0.9	69.5±0.5	63.8±0.9
QM9(zpve)	60.5±1.7	60.7±1.7	65.4±0.9	65.6±1.4	62.7±0.5	61.8±0.7	61.3±0.3	58.8±0.6	60.4±0.4	**67.5**±19.1
QM9(u0)	77.6±3.4	**79.2**±2.7	68.9±2.4	72.9±1.6	68.3±0.5	67.7±0.4	67.8±0.6	66.1±1.0	64.6±0.9	54.9±2.0
QM9(u298)	77.6±3.3	**79.1**±2.6	68.8±2.5	72.9±1.7	68.4±0.6	67.9±0.6	67.9±0.8	66.0±1.2	64.7±1.0	55.8±2.2
QM9(h298)	77.5±3.2	**79.1**±2.6	68.9±2.5	72.9±1.6	68.4±0.7	67.9±0.7	67.9±0.8	66.2±1.2	64.7±0.9	55.7±2.0
QM9(g298)	77.6±3.3	**79.2**±2.7	68.9±2.4	72.9±1.7	68.6±0.7	68.0±0.7	68.0±0.8	66.1±1.1	64.8±0.9	54.5±1.8
QM9(cv)	62.6±1.0	60.8±0.9	65.0±1.1	64.0±0.9	64.6±1.1	62.3±0.8	**65.3**±0.7	60.9±0.8	64.6±0.7	$\underline{47.5}$±9.1
MAD										
QM8(E1-CC2)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(E2-CC2)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(f1-CC2)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.03±0.00
QM8(f2-CC2)	**0.03**±0.00	**0.03**±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.03±0.00	0.06±0.01
QM8(E1-PBE0)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(E2-PBE0)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(f1-PBE0)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.03±0.00
QM8(f2-PBE0)	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.05±0.01
QM8(E1-PBE0.1)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(E2-PBE0.1)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(f1-PBE0.1)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.03±0.01
QM8(f2-PBE0.1)	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.05±0.01
QM8(E1-CAM)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.01
QM8(E2-CAM)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.02±0.00
QM8(f1-CAM)	**0.01**±0.00	**0.01**±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	0.04±0.01
QM8(f2-CAM)	**0.02**±0.00	**0.02**±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.05±0.01
QM9(mu)	0.49±0.01	0.49±0.01	0.51±0.01	0.51±0.01	**0.48**±0.00	**0.48**±0.00	**0.48**±0.00	**0.48**±0.00	**0.48**±0.00	2.01±1.24
QM9(alpha)	0.69±0.03	0.69±0.03	1.10±0.07	1.10±0.07	0.70±0.01	0.70±0.01	0.70±0.01	0.70±0.01	0.70±0.01	9.79±0.83
QM9(homo)	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	2.13±2.01
QM9(lumo)	0.00±0.00	0.00±0.00	0.01±0.00	0.01±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	5.65±3.33
QM9(gap)	0.01±0.00	0.01±0.00	0.01±0.00	0.01±0.00	**0.01**±0.00	**0.01**±0.00	**0.01**±0.00	**0.01**±0.00	**0.01**±0.00	4.34±3.35
QM9(r2)	35.56±1.29	35.56±1.29	42.12±1.56	42.12±1.56	**33.53**±0.36	**33.53**±0.36	**33.53**±0.36	**33.53**±0.36	**33.53**±0.36	188.64±9.50
QM9(zpve)	**0.00**±0.00	**0.00**±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	4.42±3.93
QM9(u0)	**1.46**±0.18	**1.46**±0.18	3.69±0.33	3.69±0.33	2.29±0.07	2.29±0.07	2.29±0.07	2.29±0.07	2.29±0.07	40.85±3.49
QM9(u298)	**1.46**±0.18	**1.46**±0.18	3.69±0.33	3.69±0.33	2.29±0.06	2.29±0.06	2.29±0.06	2.29±0.06	2.29±0.06	40.64±3.81
QM9(h298)	**1.45**±0.18	**1.45**±0.18	3.69±0.33	3.69±0.33	2.29±0.06	2.29±0.06	2.29±0.06	2.29±0.06	2.29±0.06	40.84±3.03
QM9(g298)	**1.46**±0.18	**1.46**±0.18	3.70±0.33	3.70±0.33	2.29±0.07	2.29±0.07	2.29±0.07	2.29±0.07	2.29±0.07	41.02±3.81
QM9(cv)	0.34±0.01	0.34±0.01	0.47±0.02	0.47±0.02	**0.33**±0.01	**0.33**±0.01	**0.33**±0.01	**0.33**±0.01	**0.33**±0.01	5.03±1.55

Table 12: MCR and TCR for QM8 and QM9, for 90% PI. Numbers *not* significantly lower than 90% are in bold. Like in the main text, MADSplit and CQR achieves 90% marginal coverage rate empirically as expected, but fail to cover data with more extreme responses (in the tails). Variants of LVD almost always cover empirically, measured by both MCR and TCR.

	LVD								Conformal Baselines	
	$\hat{y}^{KR}$				$\hat{y}^{NN}$					
	No-smooth		Smooth		No-smooth		Smooth			
MCR	MN	base	MN	base	MN	base	MN	$\boxed{base}$	MADSplit	CQR
QM8(E1-CC2)	**96.3**±0.7	**96.3**±0.8	**92.6**±0.9	**92.7**±0.6	**96.2**±0.8	**96.2**±0.6	**92.3**±0.6	**92.5**±0.5	**90.0**±0.5	**90.1**±0.7
QM8(E2-CC2)	**96.2**±1.0	**96.2**±0.8	**92.0**±0.5	**92.2**±0.7	**96.1**±1.0	**96.2**±0.7	**92.0**±0.5	**92.2**±0.5	**89.8**±0.6	**90.1**±0.5
QM8(f1-CC2)	**93.9**±0.9	**94.8**±0.9	**92.3**±0.7	**93.8**±0.8	**93.7**±1.0	**94.0**±1.2	**92.2**±0.8	**92.5**±1.1	**90.1**±0.7	**90.2**±0.7
QM8(f2-CC2)	**94.8**±1.3	**95.1**±1.1	**93.0**±0.6	**93.6**±0.6	**94.3**±1.3	**95.1**±1.3	**92.5**±0.8	**93.6**±0.6	**89.9**±0.7	**89.9**±0.7
QM8(E1-PBE0)	**95.9**±0.5	**96.0**±0.6	**92.3**±0.7	**92.7**±0.6	**95.7**±0.7	**95.9**±0.6	**91.9**±0.8	**92.5**±0.6	**89.7**±0.5	**90.2**±0.5
QM8(E2-PBE0)	**95.0**±0.9	**95.4**±0.9	**91.8**±0.5	**92.4**±0.8	**95.0**±0.9	**95.3**±0.9	**91.7**±0.6	**92.3**±0.6	**90.0**±0.5	**90.0**±0.6
QM8(f1-PBE0)	**93.7**±1.1	**94.3**±1.2	**92.4**±1.0	**93.1**±1.0	**93.4**±1.2	**93.9**±1.4	**91.8**±1.3	**92.4**±1.2	**90.3**±0.8	**90.2**±0.9
QM8(f2-PBE0)	**94.0**±1.8	**94.6**±1.8	**91.9**±1.1	**92.5**±0.9	**93.9**±1.9	**94.8**±1.6	**91.6**±1.2	**92.8**±0.9	**89.9**±0.9	89.7±0.5
QM8(E1-PBE0.1)	**95.8**±0.6	**96.0**±0.6	**92.2**±0.6	**92.7**±0.6	**95.6**±0.7	**95.9**±0.5	**91.9**±0.8	**92.5**±0.7	**89.8**±0.5	**89.8**±0.6
QM8(E2-PBE0.1)	**94.9**±0.8	**95.4**±0.9	**91.7**±0.4	**92.4**±0.8	**95.0**±1.0	**95.3**±0.9	**91.8**±0.5	**92.3**±0.6	**90.2**±0.5	**90.0**±0.4
QM8(f1-PBE0.1)	**93.6**±1.0	**94.3**±1.2	**92.3**±0.9	**93.1**±1.0	**93.1**±1.4	**93.9**±1.3	**91.6**±1.3	**92.3**±1.4	**89.9**±1.0	**90.0**±0.4
QM8(f2-PBE0.1)	**94.2**±1.8	**94.6**±1.8	**92.0**±1.1	**92.5**±0.9	**94.0**±1.9	**94.8**±1.7	**91.8**±1.1	**92.8**±0.9	**89.9**±0.7	**89.7**±0.8
QM8(E1-CAM)	**96.3**±1.1	**96.5**±0.9	**92.7**±0.9	**93.2**±0.8	**96.2**±1.2	**96.4**±0.9	**92.3**±0.8	**92.8**±0.6	**89.9**±0.5	**90.2**±0.4
QM8(E2-CAM)	**95.3**±1.0	**95.6**±1.1	**92.1**±0.7	**92.6**±0.8	**95.4**±0.8	**95.6**±0.9	**92.1**±0.5	**92.5**±0.7	**90.0**±0.5	**90.0**±0.6
QM8(f1-CAM)	**93.7**±1.0	**94.7**±1.2	**92.2**±0.6	**93.5**±0.8	**93.3**±0.9	**93.8**±1.2	**91.7**±0.6	**92.2**±0.7	**89.8**±0.7	**89.9**±0.7
QM8(f2-CAM)	**94.7**±1.1	**95.0**±0.8	**92.7**±0.7	**93.4**±0.6	**94.5**±1.1	**95.2**±0.9	**92.5**±0.7	**93.5**±0.7	**90.0**±0.9	**90.3**±0.8
QM9(mu)	**92.5**±1.7	**93.2**±1.6	**90.4**±0.4	**91.3**±0.5	**92.6**±1.7	**93.3**±1.5	**90.6**±0.4	**91.4**±0.4	**90.1**±0.2	**90.0**±0.3
QM9(alpha)	**94.7**±1.5	**94.6**±1.6	**90.5**±0.4	89.7±0.4	**94.5**±1.5	**94.6**±1.6	**90.1**±0.3	89.7±0.4	**90.0**±0.2	**89.9**±0.2
QM9(homo)	**92.3**±0.6	**92.5**±0.7	**90.4**±0.4	**90.4**±0.2	**92.4**±0.6	**92.6**±0.7	**90.6**±0.3	**90.7**±0.3	**90.0**±0.3	**89.9**±0.3
QM9(lumo)	**93.5**±0.9	**93.5**±1.0	**90.5**±0.3	**90.4**±0.3	**93.5**±0.9	**93.6**±0.9	**90.7**±0.4	**90.6**±0.3	**90.1**±0.3	**89.9**±0.4
QM9(gap)	**93.0**±1.2	**93.1**±1.2	**90.5**±0.3	**90.4**±0.2	**93.0**±1.1	**93.2**±1.1	**90.6**±0.3	**90.7**±0.3	**90.1**±0.2	**89.9**±0.3
QM9(r2)	**92.6**±1.1	**93.0**±1.2	**90.1**±0.3	**90.5**±0.3	**92.6**±1.2	**93.1**±1.1	**90.4**±0.4	**90.8**±0.5	**89.9**±0.4	**90.1**±0.3
QM9(zpve)	**95.3**±1.2	**95.4**±1.2	**90.5**±0.1	**90.1**±0.3	**95.2**±1.3	**95.3**±1.4	**90.3**±0.2	**89.9**±0.2	**90.0**±0.2	**90.0**±0.2
QM9(u0)	**94.9**±1.5	**94.8**±1.6	**90.9**±0.5	**90.6**±0.5	**93.3**±1.3	**93.4**±1.3	**90.1**±0.2	**90.0**±0.3	**89.9**±0.2	**90.1**±0.2
QM9(u298)	**95.1**±1.8	**95.0**±1.9	**90.9**±0.5	**90.6**±0.5	**93.6**±1.7	**93.7**±1.8	**90.0**±0.2	**89.9**±0.3	**89.9**±0.2	**90.1**±0.2
QM9(h298)	**95.0**±1.6	**94.9**±1.7	**90.9**±0.5	**90.6**±0.5	**93.5**±1.4	**93.6**±1.4	**90.1**±0.2	**90.0**±0.3	**89.9**±0.2	**90.1**±0.2
QM9(g298)	**94.9**±1.5	**94.8**±1.6	**90.8**±0.5	**90.6**±0.5	**93.3**±1.4	**93.4**±1.3	**90.0**±0.2	**89.9**±0.3	**89.9**±0.2	**90.1**±0.2
QM9(cv)	**94.3**±1.4	**94.4**±1.5	**90.4**±0.3	**89.9**±0.5	**94.2**±1.5	**94.4**±1.5	**90.4**±0.4	**90.1**±0.4	**90.0**±0.2	**90.0**±0.4
TCR										
QM8(E1-CC2)	**94.4**±1.2	**94.3**±1.4	**89.6**±1.9	89.3±0.9	**95.2**±0.9	**94.3**±1.5	**91.3**±1.2	**90.2**±1.3	88.1±1.5	83.5±6.4
QM8(E2-CC2)	**95.5**±1.5	**94.9**±1.7	**89.6**±1.2	88.6±1.4	**95.8**±1.6	**95.6**±1.5	**90.7**±1.7	**90.1**±1.7	87.4±2.2	84.4±4.4
QM8(f1-CC2)	**94.5**±2.2	**97.0**±1.7	**90.4**±1.5	**93.6**±1.4	**94.9**±1.3	**94.1**±2.0	**91.4**±1.2	**90.5**±1.8	85.0±1.8	80.1±2.8
QM8(f2-CC2)	**95.6**±2.5	**96.2**±2.7	**91.8**±1.6	**92.3**±1.7	**95.7**±2.6	**95.2**±2.9	**92.4**±1.6	**91.4**±1.5	84.9±2.1	73.3±4.8
QM8(E1-PBE0)	**94.0**±1.4	**94.2**±1.2	**89.9**±1.8	**89.9**±1.5	**94.4**±0.8	**94.2**±1.5	**90.9**±1.2	**90.3**±1.5	87.8±1.8	80.9±4.6
QM8(E2-PBE0)	**94.1**±1.6	**93.8**±2.0	**90.0**±1.6	**89.7**±2.0	**94.8**±1.4	**94.2**±1.9	**90.9**±1.7	**90.5**±2.0	87.5±1.8	82.1±5.8
QM8(f1-PBE0)	**94.3**±2.3	**97.3**±1.5	**90.9**±3.0	**94.4**±1.8	**94.5**±2.0	**95.2**±1.9	**91.1**±2.7	**91.8**±2.1	85.1±1.9	80.4±2.5
QM8(f2-PBE0)	**94.5**±3.4	**95.6**±3.4	**90.1**±2.4	**90.9**±1.5	**95.1**±3.2	**95.1**±3.6	**90.9**±2.4	**90.2**±1.8	84.4±2.6	74.6±6.2
QM8(E1-PBE0.1)	**94.0**±1.3	**94.2**±1.2	**89.8**±1.5	**89.9**±1.5	**94.3**±1.2	**93.9**±1.6	**90.7**±1.4	**90.3**±1.9	87.8±2.0	83.4±4.4
QM8(E2-PBE0.1)	**94.2**±1.4	**93.8**±2.0	**90.1**±1.6	**89.7**±2.0	**94.9**±1.5	**94.3**±2.1	**91.0**±1.5	**90.4**±2.2	87.9±1.8	82.0±5.5
QM8(f1-PBE0.1)	**94.2**±2.3	**97.3**±1.5	**90.7**±2.8	**94.4**±1.8	**93.9**±2.1	**95.2**±1.8	**90.6**±2.7	**91.7**±2.5	84.1±2.0	81.1±2.4
QM8(f2-PBE0.1)	**94.7**±3.2	**95.6**±3.4	**90.0**±2.1	**90.9**±1.5	**95.0**±3.2	**95.0**±3.7	**90.8**±2.3	**90.3**±1.9	84.2±2.4	73.3±3.8
QM8(E1-CAM)	**94.0**±1.7	**94.1**±1.7	**89.1**±2.4	**90.0**±1.3	**94.3**±1.5	**93.8**±1.9	**90.2**±1.4	**89.8**±1.7	86.5±1.4	81.4±5.8
QM8(E2-CAM)	**95.1**±1.4	**95.0**±1.8	**90.7**±1.2	**90.4**±1.5	**95.4**±1.4	**95.4**±1.9	**91.8**±1.0	**91.1**±1.5	88.1±1.7	83.5±5.0
QM8(f1-CAM)	**95.3**±2.3	**98.8**±1.2	**92.1**±1.4	**96.2**±0.7	**95.1**±1.6	**95.9**±1.4	**92.3**±0.8	**93.0**±1.3	86.7±1.1	81.9±3.3
QM8(f2-CAM)	**95.7**±1.7	**96.8**±1.8	**91.5**±0.9	**92.8**±0.9	**95.9**±1.7	**96.0**±1.9	**91.7**±1.5	**91.2**±1.5	85.0±2.0	73.5±4.6
QM9(mu)	83.2±4.7	82.9±4.6	77.4±1.5	77.1±1.7	87.0±3.5	86.0±3.4	83.1±1.5	81.9±1.1	79.8±1.5	64.7±4.2
QM9(alpha)	**96.7**±1.3	**96.7**±1.5	88.9±0.7	87.9±0.7	**97.4**±1.1	**97.5**±1.2	**90.7**±0.9	**90.3**±0.9	87.3±0.8	69.9±2.3
QM9(homo)	**91.7**±1.4	**91.4**±1.5	87.2±0.8	86.7±1.0	**93.0**±1.0	**92.9**±0.9	**89.7**±0.7	**89.4**±1.0	86.5±0.5	**89.8**±2.7
QM9(lumo)	**93.4**±1.3	**93.5**±1.2	88.2±0.8	88.1±0.7	**94.7**±0.7	**94.7**±0.8	**91.1**±0.9	**90.7**±1.0	**89.7**±0.9	87.5±1.9
QM9(gap)	**92.0**±1.8	**91.9**±2.0	87.6±0.6	86.9±0.9	**93.7**±1.4	**93.6**±1.5	**90.3**±0.8	**90.0**±1.0	87.9±0.8	88.1±1.3
QM9(r2)	**94.1**±2.1	**93.7**±2.0	88.2±0.8	88.0±1.2	**95.3**±1.7	**95.0**±1.7	**90.9**±0.7	**90.6**±1.2	87.6±0.8	67.0±4.3
QM9(zpve)	**97.0**±1.0	**97.1**±0.9	**90.4**±0.7	**90.1**±0.7	**96.7**±1.2	**96.9**±1.2	**91.2**±0.9	**90.9**±0.8	**90.5**±0.6	**88.3**±4.5
QM9(u0)	**97.3**±1.5	**97.2**±1.6	**91.2**±0.9	**90.7**±0.9	**95.9**±1.5	**96.1**±1.7	**90.7**±0.5	**90.4**±0.8	84.1±1.0	80.1±3.2
QM9(u298)	**97.4**±1.7	**97.4**±1.8	**91.2**±0.8	**90.7**±0.9	**96.2**±1.8	**96.2**±1.9	**90.6**±0.4	**90.4**±0.8	84.0±0.9	79.8±3.3
QM9(h298)	**97.4**±1.7	**97.3**±1.7	**91.2**±0.8	**90.7**±0.9	**96.1**±1.8	**96.1**±1.9	**90.7**±0.4	**90.4**±0.7	84.0±1.1	80.3±2.9
QM9(g298)	**97.3**±1.6	**97.2**±1.7	**91.1**±0.7	**90.7**±0.9	**96.0**±1.6	**96.0**±1.7	**90.7**±0.4	**90.3**±0.7	84.0±1.1	80.0±2.3
QM9(cv)	**95.9**±1.8	**95.9**±2.0	89.2±0.7	88.2±1.0	**96.5**±1.5	**96.6**±1.5	**91.0**±0.7	**90.8**±0.7	87.8±0.6	80.4±7.7

Table 13: Average width for 50% and 90% PIs of different methods. Narrowest PIs are in bold, and further underscored if significantly narrower than the second best (at $p = 0.05$). As a reminder, there are 4357 data in QM8's test set and 26744 in QM9's. Among these valid methods, LVD's 50% PIs are the narrowest, and still competitive at 90% despite satisfying a stronger coverage requirement.

		LVD								Conformal Baselines	
		$\hat{y}^{KR}$				$\hat{y}^{NN}$					
		No-smooth		Smooth		No-smooth		Smooth			
Width @ 50%	# finite	MN	base	MN	base	MN	base	MN	$\boxed{base}$	MADSplit	CQR
QM8(E1-CC2)	3791.5±189.4	1.2e-02±4.0e-04	1.1e-02±3.0e-04	1.1e-02±6.0e-04	9.8e-03±5.0e-04	1.3e-02±5.0e-04	1.2e-02±4.0e-04	1.1e-02±5.0e-04	**9.6e-03**±4.0e-04	1.0e-02±4.0e-04	3.2e-02±4.1e-03
QM8(E2-CC2)	3829.8±126.6	1.4e-02±7.0e-04	1.3e-02±5.0e-04	1.3e-02±4.0e-04	1.2e-02±4.0e-04	1.5e-02±9.0e-04	1.4e-02±7.0e-04	1.2e-02±3.0e-04	**1.1e-02**±2.0e-04	1.2e-02±3.0e-04	2.8e-02±3.3e-03
QM8(f1-CC2)	4066.0±179.7	2.1e-02±1.6e-03	1.9e-02±1.2e-03	1.8e-02±2.1e-03	**1.6e-02**±1.6e-03	2.4e-02±1.2e-03	2.1e-02±1.0e-03	1.9e-02±2.1e-03	1.6e-02±1.4e-03	1.8e-02±1.6e-03	3.0e-02±5.4e-03
QM8(f2-CC2)	3998.1±231.6	5.2e-02±3.1e-03	4.6e-02±2.5e-03	4.6e-02±4.9e-03	**4.1e-02**±4.3e-03	5.5e-02±3.2e-03	4.9e-02±3.2e-03	4.7e-02±5.1e-03	4.2e-02±4.7e-03	4.1e-02±3.2e-03	5.7e-02±1.1e-02
QM8(E1-PBE0)	3932.1±133.8	1.2e-02±3.0e-04	1.1e-02±2.0e-04	1.1e-02±6.0e-04	**9.6e-03**±3.0e-04	1.3e-02±5.0e-04	1.1e-02±4.0e-04	1.1e-02±3.0e-04	9.6e-03±2.0e-04	1.1e-02±3.0e-04	3.4e-02±9.1e-03
QM8(E2-PBE0)	4063.2±133.4	1.3e-02±4.0e-04	1.2e-02±3.0e-04	1.2e-02±3.0e-04	1.1e-02±3.0e-04	1.3e-02±6.0e-04	1.2e-02±6.0e-04	1.2e-02±3.0e-04	**1.1e-02**±3.0e-04	1.2e-02±2.0e-04	2.9e-02±7.9e-03
QM8(f1-PBE0)	4152.1±159.3	1.8e-02±9.0e-04	1.5e-02±9.0e-04	1.6e-02±2.0e-03	**1.4e-02**±1.7e-03	2.0e-02±1.0e-03	1.7e-02±1.0e-03	1.6e-02±2.3e-03	1.4e-02±2.0e-03	1.4e-02±1.6e-03	3.2e-02±6.7e-03
QM8(f2-PBE0)	4059.8±240.7	3.9e-02±1.0e-03	3.4e-02±1.0e-03	3.4e-02±3.5e-03	**3.0e-02**±2.9e-03	4.1e-02±1.6e-03	3.6e-02±1.7e-03	3.4e-02±3.5e-03	3.0e-02±2.8e-03	3.0e-02±2.3e-03	4.2e-02±5.9e-03
QM8(E1-PBE0.1)	3932.1±133.8	1.2e-02±2.0e-04	1.1e-02±2.0e-04	1.1e-02±4.0e-04	**9.6e-03**±3.0e-04	1.3e-02±5.0e-04	1.1e-02±4.0e-04	1.1e-02±4.0e-04	9.6e-03±2.0e-04	1.1e-02±3.0e-04	3.1e-02±4.8e-03
QM8(E2-PBE0.1)	4063.2±133.4	1.3e-02±4.0e-04	1.2e-02±3.0e-04	1.2e-02±3.0e-04	1.1e-02±3.0e-04	1.3e-02±5.0e-04	1.2e-02±5.0e-04	1.2e-02±3.0e-04	**1.1e-02**±3.0e-04	1.2e-02±3.0e-04	2.9e-02±3.3e-03
QM8(f1-PBE0.1)	4152.1±159.3	1.8e-02±1.1e-03	1.5e-02±9.0e-04	1.6e-02±2.0e-03	**1.4e-02**±1.7e-03	1.9e-02±1.1e-03	1.7e-02±1.0e-03	1.6e-02±2.2e-03	1.4e-02±2.0e-03	1.4e-02±1.6e-03	3.5e-02±8.0e-03
QM8(f2-PBE0.1)	4059.8±240.7	3.9e-02±1.3e-03	3.4e-02±1.0e-03	3.3e-02±3.5e-03	**3.0e-02**±2.9e-03	4.1e-02±1.9e-03	3.6e-02±1.8e-03	3.4e-02±3.4e-03	3.0e-02±3.0e-03	3.0e-02±2.5e-03	4.3e-02±7.6e-03
QM8(E1-CAM)	3782.7±252.3	1.1e-02±4.0e-04	1.0e-02±3.0e-04	1.0e-02±3.0e-04	**9.1e-03**±2.0e-04	1.2e-02±5.0e-04	1.1e-02±6.0e-04	1.0e-02±4.0e-04	9.3e-03±3.0e-04	1.0e-02±3.0e-04	2.9e-02±3.1e-03
QM8(E2-CAM)	4004.4±151.5	1.2e-02±3.0e-04	1.1e-02±3.0e-04	1.1e-02±3.0e-04	1.0e-02±3.0e-04	1.3e-02±6.0e-04	1.2e-02±7.0e-04	1.1e-02±3.0e-04	**1.0e-02**±3.0e-04	1.1e-02±3.0e-04	2.7e-02±3.2e-03
QM8(f1-CAM)	4125.5±161.7	1.9e-02±1.2e-03	1.6e-02±1.1e-03	1.6e-02±2.0e-03	**1.4e-02**±1.7e-03	2.2e-02±1.0e-03	1.9e-02±1.0e-03	1.8e-02±1.4e-03	1.6e-02±1.2e-03	1.6e-02±9.0e-04	3.1e-02±4.1e-03
QM8(f2-CAM)	4001.1±224.0	4.1e-02±1.8e-03	3.7e-02±1.3e-03	3.6e-02±3.0e-03	**3.2e-02**±2.6e-03	4.4e-02±1.2e-03	4.0e-02±1.3e-03	3.8e-02±3.0e-03	3.3e-02±2.8e-03	3.3e-02±2.3e-03	5.1e-02±5.8e-03
QM9(mu)	25845.8±660.3	8.9e-01±5.7e-02	8.1e-01±3.1e-02	8.4e-01±2.7e-02	7.6e-01±1.9e-02	8.7e-01±5.3e-02	7.8e-01±4.2e-02	7.9e-01±2.1e-02	**7.1e-01**±1.7e-02	7.7e-01±2.0e-02	2.1e+00±4.2e-01
QM9(alpha)	24665.0±1191.1	1.2e+00±3.3e-02	1.1e+00±4.1e-02	1.6e+00±1.3e-01	1.4e+00±1.2e-01	1.2e+00±3.2e-02	1.1e+00±4.8e-02	9.9e-01±4.2e-02	**8.8e-01**±2.2e-02	1.0e+00±3.3e-02	1.4e+01±8.4e-01
QM9(homo)	26024.3±221.8	7.5e-03±3.0e-04	7.1e-03±3.0e-04	7.4e-03±2.0e-04	7.0e-03±2.0e-04	7.0e-03±1.0e-04	6.6e-03±1.0e-04	6.6e-03±1.0e-04	**6.2e-03**±1.0e-04	6.6e-03±1.0e-04	2.2e+00±1.7e+00
QM9(lumo)	25608.8±416.6	8.5e-03±2.0e-04	8.1e-03±2.0e-04	8.3e-03±2.0e-04	7.9e-03±3.0e-04	7.9e-03±2.0e-04	7.5e-03±2.0e-04	7.2e-03±1.0e-04	**6.8e-03**±1.0e-04	7.2e-03±1.0e-04	2.1e+00±1.5e+00
QM9(gap)	25862.9±476.1	1.1e-02±2.0e-04	1.0e-02±2.0e-04	1.1e-02±4.0e-04	9.8e-03±3.0e-04	1.0e-02±5.0e-04	9.6e-03±6.0e-04	9.5e-03±2.0e-04	**8.8e-03**±3.0e-04	9.4e-03±2.0e-04	1.1e+00±5.9e-01
QM9(r2)	25736.3±537.4	6.3e+01±2.0e+00	5.7e+01±1.6e+00	6.9e+01±2.5e+00	6.2e+01±2.7e+00	5.8e+01±1.9e+00	5.2e+01±1.9e+00	5.4e+01±7.8e-01	**4.8e+01**±6.1e-01	5.4e+01±9.5e-01	2.4e+02±1.1e+01
QM9(zpve)	24486.6±870.8	1.6e-03±2.0e-04	**1.5e-03**±1.0e-04	3.8e-03±4.0e-04	3.6e-03±4.0e-04	2.8e-03±1.0e-04	2.7e-03±1.0e-04	2.4e-03±1.0e-04	2.2e-03±1.0e-04	2.4e-03±1.0e-04	1.9e+00±1.6e+00
QM9(u0)	25375.4±499.9	1.9e+00±2.4e-01	**1.7e+00**±2.1e-01	4.9e+00±6.5e-01	4.2e+00±5.2e-01	3.7e+00±1.4e-01	3.2e+00±1.4e-01	3.0e+00±1.3e-01	2.5e+00±9.2e-02	2.9e+00±1.3e-01	4.4e+01±3.7e+00
QM9(u298)	25204.3±754.3	1.9e+00±2.5e-01	**1.7e+00**±2.0e-01	4.9e+00±6.9e-01	4.2e+00±5.3e-01	3.7e+00±1.3e-01	3.2e+00±1.5e-01	3.0e+00±1.5e-01	2.5e+00±8.5e-02	2.9e+00±1.3e-01	4.5e+01±3.7e+00
QM9(h298)	25288.9±581.3	1.9e+00±2.4e-01	**1.7e+00**±2.0e-01	4.9e+00±6.8e-01	4.2e+00±5.3e-01	3.7e+00±1.3e-01	3.2e+00±1.4e-01	3.0e+00±1.3e-01	2.5e+00±8.1e-02	2.9e+00±1.1e-01	4.4e+01±3.8e+00
QM9(g298)	25345.2±543.2	1.9e+00±2.5e-01	**1.7e+00**±2.1e-01	4.9e+00±6.5e-01	4.2e+00±5.2e-01	3.7e+00±1.3e-01	3.2e+00±1.4e-01	3.1e+00±1.3e-01	2.5e+00±8.8e-02	2.9e+00±1.3e-01	4.4e+01±3.7e+00
QM9(cv)	25051.2±829.4	6.0e-01±2.1e-02	5.5e-01±2.0e-02	6.9e-01±5.5e-02	6.3e-01±4.7e-02	5.7e-01±1.8e-02	5.3e-01±2.0e-02	4.9e-01±1.5e-02	**4.4e-01**±8.6e-03	4.9e-01±1.5e-02	5.2e+00±1.3e+00
Width @ 90%											
QM8(E1-CC2)	2373.2±351.0	2.8e-02±2.1e-03	2.7e-02±8.0e-04	2.6e-02±1.4e-03	2.5e-02±9.0e-04	2.9e-02±2.6e-03	2.7e-02±8.0e-04	2.5e-02±1.2e-03	**2.4e-02**±7.0e-04	2.5e-02±1.4e-03	8.4e-02±1.4e-02
QM8(E2-CC2)	2356.4±368.3	3.3e-02±1.2e-03	3.3e-02±8.0e-04	3.0e-02±1.2e-03	2.9e-02±1.0e-03	3.3e-02±1.2e-03	3.3e-02±1.0e-03	2.9e-02±2.0e-04	2.9e-02±5.0e-04	**2.8e-02**±6.0e-04	8.8e-02±1.5e-02
QM8(f1-CC2)	3239.5±414.0	4.4e-02±7.7e-03	3.2e-02±5.9e-03	3.6e-02±7.0e-03	**2.9e-02**±5.0e-03	5.6e-02±8.8e-03	3.6e-02±6.1e-03	4.0e-02±8.0e-03	3.2e-02±5.4e-03	4.0e-02±6.9e-03	8.8e-02±9.8e-03
QM8(f2-CC2)	3061.2±504.8	1.1e-01±1.5e-02	8.6e-02±1.3e-02	9.3e-02±1.8e-02	**7.7e-02**±1.5e-02	1.1e-01±1.9e-02	8.8e-02±1.3e-02	9.2e-02±1.7e-02	8.1e-02±1.4e-02	8.5e-02±1.2e-02	1.5e-01±3.2e-02
QM8(E1-PBE0)	2557.4±218.2	2.9e-02±2.1e-03	2.8e-02±8.0e-04	2.6e-02±1.5e-03	2.6e-02±8.0e-04	2.9e-02±2.6e-03	2.8e-02±9.0e-04	**2.5e-02**±1.2e-03	2.6e-02±8.0e-04	2.6e-02±1.1e-03	9.1e-02±5.9e-03
QM8(E2-PBE0)	2956.5±408.3	3.2e-02±1.0e-03	3.2e-02±7.0e-04	2.9e-02±1.1e-03	2.9e-02±1.0e-03	3.2e-02±9.0e-04	3.2e-02±6.0e-04	2.8e-02±9.0e-04	2.9e-02±7.0e-04	**2.8e-02**±4.0e-04	7.5e-02±7.2e-03
QM8(f1-PBE0)	3493.5±488.2	3.6e-02±6.8e-03	2.8e-02±6.9e-03	3.3e-02±7.6e-03	**2.6e-02**±7.3e-03	4.0e-02±6.0e-03	3.1e-02±6.9e-03	3.4e-02±8.7e-03	3.0e-02±7.2e-03	3.2e-02±5.3e-03	9.1e-02±1.3e-02
QM8(f2-PBE0)	3155.5±646.3	8.4e-02±1.8e-02	6.8e-02±1.2e-02	7.1e-02±1.9e-02	**5.9e-02**±1.4e-02	8.3e-02±1.6e-02	6.9e-02±1.2e-02	6.9e-02±1.8e-02	6.2e-02±1.4e-02	6.8e-02±1.3e-02	1.3e-01±2.6e-02
QM8(E1-PBE0.1)	2557.4±218.2	3.0e-02±5.1e-03	2.8e-02±8.0e-04	2.6e-02±1.5e-03	2.6e-02±8.0e-04	2.9e-02±2.7e-03	2.8e-02±9.0e-04	**2.5e-02**±1.2e-03	2.6e-02±7.0e-04	2.6e-02±1.1e-03	8.1e-02±1.2e-02
QM8(E2-PBE0.1)	2956.5±408.3	3.2e-02±9.0e-04	3.2e-02±7.0e-04	2.9e-02±1.1e-03	2.9e-02±1.0e-03	3.2e-02±9.0e-04	3.2e-02±6.0e-04	2.9e-02±9.0e-04	2.9e-02±7.0e-04	**2.8e-02**±5.0e-04	8.4e-02±7.6e-03
QM8(f1-PBE0.1)	3493.5±488.2	3.5e-02±8.4e-03	2.8e-02±6.9e-03	3.3e-02±7.9e-03	**2.6e-02**±7.3e-03	3.8e-02±7.5e-03	3.1e-02±6.9e-03	3.4e-02±8.3e-03	2.9e-02±7.2e-03	3.2e-02±5.6e-03	8.9e-02±1.7e-02
QM8(f2-PBE0.1)	3155.5±646.3	8.4e-02±2.1e-02	6.8e-02±1.2e-02	7.2e-02±2.0e-02	**5.9e-02**±1.4e-02	8.3e-02±1.7e-02	6.9e-02±1.1e-02	6.9e-02±1.8e-02	6.1e-02±1.4e-02	6.8e-02±1.3e-02	1.2e-01±2.0e-02
QM8(E1-CAM)	2317.5±472.6	2.8e-02±2.3e-03	2.6e-02±9.0e-04	2.5e-02±1.1e-03	2.4e-02±7.0e-04	2.8e-02±2.0e-03	2.7e-02±9.0e-04	2.4e-02±1.2e-03	2.4e-02±8.0e-04	**2.4e-02**±1.1e-03	9.0e-02±2.5e-02
QM8(E2-CAM)	2791.8±491.5	2.9e-02±1.0e-03	3.0e-02±7.0e-04	2.7e-02±1.2e-03	2.7e-02±9.0e-04	2.9e-02±1.0e-03	3.0e-02±8.0e-04	2.7e-02±1.2e-03	2.7e-02±8.0e-04	**2.6e-02**±6.0e-04	7.6e-02±1.0e-02
QM8(f1-CAM)	3366.6±448.1	3.4e-02±9.3e-03	2.8e-02±7.5e-03	3.0e-02±8.9e-03	**2.5e-02**±7.2e-03	4.3e-02±9.2e-03	3.2e-02±7.0e-03	3.4e-02±1.0e-02	2.9e-02±6.9e-03	3.4e-02±8.1e-03	1.0e-01±2.5e-02
QM8(f2-CAM)	3059.9±482.0	8.8e-02±1.6e-02	7.0e-02±1.3e-02	7.4e-02±1.6e-02	**6.2e-02**±1.4e-02	9.0e-02±1.4e-02	7.1e-02±1.3e-02	7.3e-02±1.7e-02	6.4e-02±1.3e-02	6.9e-02±1.3e-02	1.4e-01±1.3e-02
QM9(mu)	23224.8±2625.8	2.2e+00±6.9e-02	2.2e+00±5.7e-02	2.1e+00±5.7e-02	2.1e+00±4.1e-02	2.1e+00±1.1e-01	2.2e+00±7.5e-02	**1.9e+00**±4.8e-02	2.1e+00±3.7e-02	2.0e+00±3.7e-02	7.1e+00±2.8e+00
QM9(alpha)	19039.0±3410.6	2.7e+00±1.1e-01	2.6e+00±7.5e-02	3.5e+00±3.6e-01	3.4e+00±2.8e-01	2.6e+00±9.3e-02	2.5e+00±7.7e-02	2.3e+00±1.2e-01	**2.3e+00**±7.0e-02	2.4e+00±9.0e-02	4.0e+01±2.7e+00
QM9(homo)	23865.0±834.2	1.8e-02±6.0e-04	1.8e-02±4.0e-04	1.8e-02±4.0e-04	1.8e-02±3.0e-04	1.7e-02±3.0e-04	1.8e-02±3.0e-04	**1.7e-02**±3.0e-04	1.7e-02±1.0e-04	1.7e-02±2.0e-04	6.7e+00±3.8e+00
QM9(lumo)	21867.7±1906.7	2.1e-02±5.0e-04	2.1e-02±5.0e-04	2.0e-02±5.0e-04	2.0e-02±5.0e-04	1.9e-02±4.0e-04	1.9e-02±5.0e-04	**1.8e-02**±3.0e-04	1.8e-02±2.0e-04	1.8e-02±2.0e-04	1.3e+01±7.1e+00
QM9(gap)	22868.0±2105.0	2.7e-02±7.0e-04	2.7e-02±8.0e-04	2.6e-02±7.0e-04	2.5e-02±6.0e-04	2.5e-02±1.0e-03	2.5e-02±1.1e-03	**2.3e-02**±4.0e-04	2.3e-02±5.0e-04	2.4e-02±4.0e-04	1.1e+01±6.1e+00
QM9(r2)	22850.8±1979.8	1.5e+02±4.8e+00	1.6e+02±3.4e+00	1.6e+02±8.0e+00	1.6e+02±6.6e+00	1.4e+02±2.6e+00	1.5e+02±3.0e+00	**1.3e+02**±2.7e+00	1.4e+02±2.4e+00	1.3e+02±2.5e+00	7.6e+02±2.9e+01
QM9(zpve)	17884.2±2703.3	3.6e-03±5.0e-04	**3.5e-03**±4.0e-04	8.8e-03±9.0e-04	8.7e-03±8.0e-04	6.0e-03±2.0e-04	6.0e-03±2.0e-04	5.6e-03±2.0e-04	5.6e-03±2.0e-04	5.8e-03±2.0e-04	1.1e+01±7.8e+00
QM9(u0)	21369.4±1980.2	5.3e+00±9.9e-01	**4.6e+00**±7.7e-01	1.2e+01±1.4e+00	1.0e+01±1.3e+00	7.8e+00±5.4e-01	7.0e+00±3.1e-01	6.8e+00±5.5e-01	6.4e+00±4.6e-01	7.6e+00±3.0e-01	1.7e+02±1.5e+01
QM9(u298)	20728.2±2926.0	5.3e+00±1.0e+00	**4.6e+00**±7.8e-01	1.2e+01±1.6e+00	1.0e+01±1.4e+00	7.8e+00±5.5e-01	7.1e+00±3.0e-01	6.8e+00±6.1e-01	6.4e+00±5.1e-01	7.5e+00±3.1e-01	1.7e+02±1.5e+01
QM9(h298)	21053.4±2274.2	5.2e+00±1.0e+00	**4.6e+00**±8.1e-01	1.2e+01±1.5e+00	1.0e+01±1.4e+00	7.8e+00±5.6e-01	7.0e+00±3.4e-01	6.8e+00±5.6e-01	6.4e+00±4.8e-01	7.5e+00±2.9e-01	1.7e+02±1.3e+01
QM9(g298)	21269.7±2092.8	5.3e+00±9.8e-01	**4.6e+00**±7.7e-01	1.2e+01±1.4e+00	1.0e+01±1.3e+00	7.8e+00±5.4e-01	7.0e+00±3.1e-01	6.8e+00±5.5e-01	6.4e+00±4.6e-01	7.6e+00±3.0e-01	1.7e+02±1.5e+01
QM9(cv)	20084.8±2942.8	1.4e+00±6.2e-02	1.4e+00±6.0e-02	1.6e+00±1.5e-01	1.5e+00±1.1e-01	1.3e+00±2.5e-02	1.3e+00±3.4e-02	1.2e+00±3.7e-02	**1.2e+00**±2.8e-02	1.2e+00±3.3e-02	2.1e+01±5.5e+00

Table 14: (On QM9) MCR conditioning on the presence of certain functional groups in the original SMILES representation. The list of groups are taken from the OPENSMILES project, as implemented in [32]. The (pooled) mean coverage rate are computed over 10 random re-splits of the full QM9 dataset like in other parts of this paper. We keep only the functional groups with at least 200 appearances in all ten randomly sampled test set. All numbers not significantly lower than 90% at $p = 0.05$ are in bold.

Conditional Coverage Rate	alpha	cv	g298	gap	h298	homo	lumo	mu	r2	u0	u298	zpve
2-butyne	88.4±0.4	**94.6**±0.4	**93.1**±0.4	**94.9**±0.4	**92.8**±0.4	**95.6**±0.4	**94.4**±0.4	**94.3**±0.4	**93.2**±0.4	**92.8**±0.4	**93.0**±0.4	**92.2**±0.4
aldehyde	**93.3**±0.4	**94.4**±0.4	**94.8**±0.4	**92.6**±0.4	**94.4**±0.4	**94.7**±0.4	**91.1**±0.4	**90.7**±0.4	**90.4**±0.4	**94.4**±0.4	**94.6**±0.4	**93.6**±0.4
amide	**89.7**±0.3	**89.7**±0.3	**90.0**±0.3	**91.1**±0.3	**90.6**±0.3	**92.2**±0.3	**89.9**±0.3	88.0±0.3	**91.4**±0.3	**90.3**±0.3	**90.2**±0.3	**90.3**±0.3
carboxylic acid	**91.9**±0.4	**93.2**±0.4	**90.0**±0.4	**91.4**±0.4	**89.7**±0.4	**92.3**±0.4	**92.4**±0.4	**92.7**±0.4	**91.8**±0.4	**90.0**±0.4	**90.0**±0.4	**91.1**±0.4
cyclopropane	**89.9**±0.3	**90.5**±0.3	**93.5**±0.3	**90.9**±0.3	**93.7**±0.3	**89.8**±0.3	**91.8**±0.3	**93.1**±0.3	88.7±0.3	**93.6**±0.3	**93.8**±0.3	**91.2**±0.3
dimethyl ether	**90.4**±0.1	**91.9**±0.1	89.4±0.1	**91.6**±0.1	89.3±0.1	**91.9**±0.1	**91.5**±0.1	**92.2**±0.1	88.7±0.1	89.3±0.1	89.4±0.1	**90.6**±0.1
ester	**92.5**±0.6	**94.0**±0.6	**90.8**±0.6	**90.2**±0.6	**90.2**±0.6	**91.5**±0.6	**91.4**±0.6	**91.9**±0.6	**92.3**±0.6	**90.3**±0.6	**90.6**±0.6	**92.3**±0.6
ethanol	**90.0**±0.2	**92.0**±0.2	**90.0**±0.2	**93.1**±0.2	**90.1**±0.2	**92.9**±0.2	**92.5**±0.2	**91.8**±0.2	88.1±0.2	**90.0**±0.2	**90.1**±0.2	**90.8**±0.2
ethene	84.1±0.2	89.5±0.2	86.7±0.2	**90.3**±0.2	86.9±0.2	**91.4**±0.2	89.3±0.2	**92.8**±0.2	**90.0**±0.2	86.7±0.2	86.8±0.2	87.0±0.2
ether	**90.4**±0.1	**91.9**±0.1	89.4±0.1	**91.6**±0.1	89.3±0.1	**91.9**±0.1	**91.5**±0.1	**92.2**±0.1	88.7±0.1	89.3±0.1	89.4±0.1	**90.6**±0.1
formaldehyde	**90.3**±0.2	**91.5**±0.2	**91.9**±0.2	**90.9**±0.2	**92.0**±0.2	**91.7**±0.2	**89.7**±0.2	86.7±0.2	88.3±0.2	**91.9**±0.2	**91.9**±0.2	**91.5**±0.2
hydrogen cyanide	**93.8**±0.2	**93.7**±0.2	**93.4**±0.2	**89.9**±0.2	**93.4**±0.2	89.6±0.2	**92.1**±0.2	88.3±0.2	**92.2**±0.2	**93.3**±0.2	**93.1**±0.2	**91.6**±0.2
ketone	88.7±0.3	**90.9**±0.3	89.4±0.3	**92.2**±0.3	89.5±0.3	**91.5**±0.3	**90.7**±0.3	**90.6**±0.3	**90.7**±0.3	**89.7**±0.3	**89.8**±0.3	**89.8**±0.3
prop-1-ene	84.5±0.3	**89.6**±0.3	85.2±0.3	**90.4**±0.3	85.6±0.3	**92.5**±0.3	89.3±0.3	**94.1**±0.3	89.3±0.3	85.3±0.3	85.5±0.3	86.1±0.3
prop-1-yne	**89.8**±0.3	**94.8**±0.3	**94.0**±0.3	**94.6**±0.3	**93.6**±0.3	**95.4**±0.3	**94.0**±0.3	**94.4**±0.3	**92.2**±0.3	**93.8**±0.3	**93.8**±0.3	**92.8**±0.3

C Discussion on Discriminative Jackknife

Discriminative Jackknife (DJ) was recently proposed as a post-hoc method to construct prediction intervals for regression deep learning models [1]. [1] claims that DJ is simultaneously marginally valid and discriminative. Unfortunately, *neither claim is true*, and it has other practical issues, as we will discuss in detail in this section.

C.1 Jackknife+ vs. DJ

Although it is out-of-scope for this paper, we would like to briefly explain where the finite-sample coverage guarantee comes from, or rather *should have* come from. It is highly recommended that the readers read the original work of Jackknife+, [4] which lays the theoretical foundation for [1] more details.

Suppose we have training data $\{Z_i\}_{i=1}^n$ where $Z_i = (X_i, Y_i)$, and $(X, Y) \sim \mathcal{P}$ for some unknown distribution $\mathcal{P}$. Suppose we have an *order-invariant* algorithm $\mathcal{A}$ that trains a mean-estimator given some data. We will denote the full estimator as $\hat{\mu}$, the leave-one-out (LOO) estimator as $\hat{\mu}_{-i}$, and the LOO residual as R_i^{LOO}, defined as:

$$\hat{\mu} := \mathcal{A}\big(\{(X_j, Y_j)\}_{j\in[n]}\big) \tag{24}$$

$$\hat{\mu}_{-i} := \mathcal{A}\big(\{(X_j, Y_j)\}_{j\in[n]\setminus\{i\}}\big) \tag{25}$$

$$R_i^{LOO} := |Y_i - \hat{\mu}_{-i}(X_i)| \tag{26}$$

We will also define $\hat{q}_{n,\beta}^{+}\{v_i\}$ as the $\lceil (n+1)\beta \rceil$-th smallest (close to the β-th quantile) of $v_1, \ldots, v_n$, and $\hat{q}_{n,\beta}^{-}\{v_i\}$ as the $\lfloor (n+1)\beta \rfloor$ smallest value[10].

The original Jackknife+ [4] does the following to construct a PI with finite-sample coverage guarantee (at level $1 - 2\alpha$, but empirically usually covers $1 - \alpha$ of the time):

- Step 1: Train the LOO estimator $\hat{\mu}_{-i}$ for $i \in [n]$.
- Step 2: Collect the LOO residuals R_i^{LOO} for $i \in [n]$.
- Step 3 (inference): For a new data point (X_{n+1}, Y_{n+1}), the Jackknife+ PI would be

$$\hat{C}_\alpha^{Jackknife+}(X_{n+1}) := [\hat{q}_{n,\alpha}^{-}\{\hat{\mu}_{-i}(X_{n+1}) - R_i^{LOO}\}, \hat{q}_{n,1-\alpha}^{+}\{\hat{\mu}_{-i}(X_{n+1}) + R_i^{LOO}\}] \tag{27}$$

Assuming exchangeability of $\{Z_i\}_{i=1}^{n+1}$, [4] proves that

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_\alpha^{Jackknife+}(X_{n+1})\} \geq 1 - 2\alpha \tag{28}$$

Here the probability is taken over all training samples and the test data.

DJ aims to apply the above for deep learning algorithm $\mathcal{A}$. The only difference between DJ and Jackknife+ is that replaces step 1 with step 1b below:

- Step 1b: replace $\hat{\mu}_{-i}$ with $\hat{\mu}_{-i}^{HOIF}$, which is estimated using $\hat{\mu}$ and higher-order influence function (HOIF) without actually retraining the deep learning algorithm $\mathcal{A}$.

C.2 Validity

Although using influence function (IF) to estimate $\hat{\mu}_{-i}$ is possible, in practice, there is almost no way to do this. For the coverage guarantee (Theorem 1 in [4]) to hold, it is important that for $\hat{\mu}_{-i}$, Z_i and Z_{n+1} are also "exchangeable". In other words, $\hat{\mu}_{-i}$ cannot see Z_i at all, which is crucial in Step 2 of the proof of Theorem 1 (Section 6 in [4]). If $\hat{\mu}_{-i}$ actually "remember" Z_i somehow, then the last step of Step 2 in the proof breaks.

Unfortunately, $\hat{\mu}_{-i}^{HOIF}$ *does* "remember" the Z_i it saw. The original paper [18] also uses IF to estimate the LOO models, but it only applies this to understand which training sample has more influence on the model, or in some qualitative assessment settings (as the name of the paper suggests). Even for such use case, [6] summarizes several issues with using IF in deep learning, one of which

---

[10] Note the $+$ and $-$ signs are used to distinguish the $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ operations.

is the error in estimating just the ranking of the influences even with first order IF estimated with *exact* inverse-Hessian vector product (HVP). In [1], the HOIFs are computed recursively, and every IF is computed with *approximate* HVP, which means there is little understanding in the quality of such estimates[11]. To actually achieve the theoretical guarantee in this setting, we need to eliminate completely the influence of Z_i on the model parameters of $\hat{\mu}$, which requires infinite-order exact IF and is clearly unrealistic.

C.3 Discrimination

The short answer to this is DJ is actually not discriminative, or at least not in practice. This can be found in our experiments in Section 4.2. [1] reports high AUPRC due to a code error[12]. It is worth noting that the exact version of Jackknife+ does not show discrimination in the way claimed in [1] either (See the comparison in Figure 3). The varying width of the PI is a by-product of the construction and proof, and is usually close to constant in practice. Intuitively, as $n \to \infty$, $\hat{\mu}_{-i} \to \hat{\mu}$, and the variance of the PI width would $\to 0$. Here are two simple thought experiments:

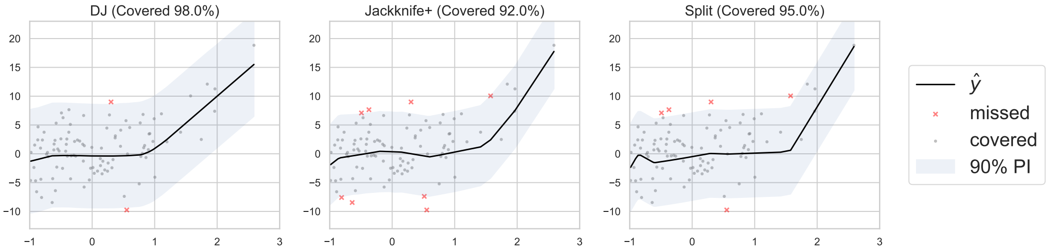

Figure 3: DJ, Jackknife+ and Split conformal on the synthetic dataset.

1. As $n \to \infty$, $\hat{\mu}_{-i} \to \hat{\mu}$, and the variance of the PI width would $\to 0$.
2. Suppose X follows a uniform distribution from $\{-10, -9, \ldots, 0, \ldots, 9, 10\}$, and $Y = |X|$. Suppose $\mathcal{A}$ is a linear regression algorithm without intercept. As long as n is big enough, the PI for any input X_{n+1} would be $[-9, 9]$. The error would however $\to |X_{n+1}|$ (because $\hat{\mu}(x) \to 0$), so there is not discrimination at all.

As a result, DJ, the approximated version, could only potentially be discriminative due to some numerical instability and/or some effect that is orthogonal to the LOO procedure and the construction of the PI, which requires more exploration and detailed explanation.

C.4 Other Considerations

Order-invariance for $\mathcal{A}$ is rarely satisfied for the deep learning model. This is because a deep learning model usually uses some variants of stochastic gradient descent (SGD) instead of gradient descent, which means permuting the input data would result in different $\hat{\mu}$. However, it is also required for the proof in [4]. [1] did not mention this at all, which results in an incomplete proof even if every stated above is fixed. That said, the proof [4] could easily be extended to training DNN with SGD as well; however it is out of the scope of this discussion.

Scalability of the proposed method in DJ is not practical, even the employed approximations. At training time, at least for the experiments in [1], directly performing the LOO procedure is faster than actually computing influence functions and estimating $\hat{\mu}_{-i}$. This of course depends on the number of training data points vs. the number of parameters of the DNN. However, as we will discuss in Section C.5, there is no strong argument for using DJ in any scenario. Moreover, if we do not store all the LOO model weights (which has a large space requirement), we would need to compute the IFs on the fly for each test data, which is prohibitively expensive.

Stability is another concern. In using the influence function, inverting Hessian is very expensive, so DJ follows [18] in using a stochastic Hessian Vector Product (HVP) method. However, one would

---

[11] In fact, based on the experiment, the errors seem to build up, as we will discuss in Section C.3.

[12] `https://github.com/ahmedmalaa/discriminative-jackknife/blob/e012d0a359aa8dac16fe03a99fa586966cf86ffe/UCI_experiments.py#L82`

also need to get a good estimate of the eigenvalue of the Hessian[13] for the HVP estimation process to converge meaningfully. In our experiment (and in the code published by the authors of [1]), exact Hessian with small NNs have to be used, instead of HVP, due to stability issues.

C.5 Conclusion

If we take a step back, Jackknife+ was proposed as an improved version of the classical Jackknife with a finite-sample marginal coverage guarantee. The question it tries to address however is not just concerning finite-sample marginal coverage, but also about data scarcity: As noted in the original Jackknife+ paper [4], split conformal already has a finite-sample guarantee (at $1-\alpha$ level as opposed to $1-2\alpha$ of Jackknife+), but the limitation is that it requires reserving a hold-out set. When the model requires more data to train, this might result in a poor fit. Of course, it is desirable to use all the data we have to train the base model. However, in many cases we only need a small portion of the data as the validation/calibration set. If data is abundant, this is not a concern, so one could use split conformal (or CQR, MADSplit, LVD, etc.). If the data is actually very scarce, then usually the model cannot be too complicated, so directly performing the LOO cross-validation with Jackknife+ would not be too expensive and will keep the theoretical guarantee. If we use DJ, we might spend more time while breaking the theoretical guarantee.

---

[13] which can be very large and thus unstable to estimate according to [6]

D Data

In this section, we will try our best to list the licenses of the public datasets we use and details about how the consent was obtained.

- UCI Yacht Hydrodynamics (Yacht)[38]: We could not find the license for this dataset. The dataset was created by "Ship Hydromechanics Laboratory, Maritime and Transport Technology Department, Technical University of Delft", and donated by "Dr Roberto Lopez" per [38].
- UCI Bikesharing (Bike) [35, 10]: The original data was provided according to the Capital Bikeshare Data License Agreement `https://www.capitalbikeshare.com/data-license-agreement`. We could not find details on how the data was obtained.
- UCI Energy Efficiency (Energy)[37, 34]: We could not find the license for this dataset. The dataset was created by Angeliki Xifara (angxifara '@' gmail.com, Civil/Structural Engineer) and was processed by Athanasios Tsanas (tsanasthanasis '@' gmail.com, Oxford Centre for Industrial and Applied Mathematics, University of Oxford, UK).
- UCI Concrete Compressive Strength (Concrete)[36, 46]: We could not find the license for this dataset. The dataset was original owned and donated by Prof. I-Cheng Yeh at Department of Information Management at Chung-Hua University, Taiwan, R.O.C.
- Boston Housing (Housing)[9]: We could not find the license for this dataset. This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass[14].
- Kin8nm[16]: We could not find the license for this dataset. The original parent dataset (the "kin" dataset) was contributed by Zoubin Ghahramani[15].
- QM8 [28, 30] and QM9 [30, 27]: We could not find the original license for these datasets, but they are discributed under CC By 4.0 [16]. They are obtained in [28] and [27].

---

[14] https://www.cs.toronto.edu/ delve/data/boston/bostonDetail.html

[15] https://www.cs.toronto.edu/ delve/data/kin/desc.html

[16] https://tdcommons.ai/single_pred_tasks/qm/