ACKNOWLEDGMENTS

This work was supported by NSF award SCH-2205289, SCH-2014438, IIS-1838042, NIH award R01 1R01NS107291-01.

REFERENCES

Siddharth Biswal, Joshua A. Kulas, Haoqi Sun, Balaji Goparaju, Michael Brandon Westover, Matt T. Bianchi, and Jimeng Sun. Sleepnet: Automated sleep staging system via deep learning. *ArXiv*, abs/1707.08262, 2017.

Glenn W. Brier. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, 78(1):1, January 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

José E. Chacón and Tarn Duong. *Multivariate Kernel Smoothing and its Applications*. Chapman and Hall, 2018.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pp. 109–116, Arlington, Virginia, USA, 2010. AUAI Press. ISBN 9780974903965.

Gari D. Clifford, Chengyu Liu, Benjamin Moody, Liwei H. Lehman, Ikaro Silva, Qiao Li, A. E. Johnson, and Roger G. Mark. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In *Computing in Cardiology*, 2017. doi: 10.22489/CinC.2017.065-469.

Morris H. Degroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8599–8603, 2013. doi: 10.1109/ICASSP.2013.6639344.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

ATelmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. Classifier calibration: How to assess and improve predicted class probabilities: a survey. *CoRR*, abs/2112.10327, 2021. URL `https://arxiv.org/abs/2112.10327`.

Sébastien Gadat, Thierry Klein, and Clément Marteau. Classification in general finite dimensional spaces with the k-nearest neighbor rule. *The Annals of Statistics*, 44(3):982–1009, 2016. ISSN 00905364. URL `http://www.jstor.org/stable/43818918`.

Wendong Ge, Jin Jing, Sungtae An, Aline Herlopian, Marcus Ng, Aaron F. Struck, Brian Appavu, Emily L. Johnson, Gamaleldin Osman, Hiba A. Haider, Ioannis Karakis, Jennifer A. Kim, Jonathan J. Halford, Monica B. Dhakar, Rani A. Sarkis, Christa B. Swisher, Sarah Schmitt, Jong Woo Lee, Mohammad Tabaeizadeh, Andres Rodriguez, Nicolas Gaspard, Emily Gilmore, Susan T. Herman, Peter W. Kaplan, Jay Pathmanathan, Shenda Hong, Eric S. Rosenthal, Sahar Zafar, Jimeng Sun, and M. Brandon Westover. Deep active learning for interictal ictal injury continuum EEG patterns. *Journal of Neuroscience Methods*, 351:108966, mar 2021. doi: 10.1016/j.jneumeth.2020.108966. URL `https://doi.org/10.1016%2Fj.jneumeth.2020.108966`.

A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 2000. ISSN 15244539. doi: 10.1161/01.cir.101.23.e215.

Alexander G. Gray and Andrew W. Moore. Nonparametric density estimation: Toward computational tractability. In *SDM*, 2003.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Chirag Gupta and Aaditya Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3942–3952. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/gupta21b.html.

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eQe8DEWNN2W.

Shenda Hong, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. Mina: Multilevel knowledge-guided attention for modeling electrocardiography signals. In *IJCAI International Joint Conference on Artificial Intelligence*, 2019. ISBN 9780999241141. doi: 10.24963/ijcai.2019/816.

Heinrich Jiang. Uniform convergence rates for kernel density estimation. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1694–1703. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jiang17b.html.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/7180cffd6a8e829dacfc2a31b3f72ece-Paper.pdf.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *J. Am. Medical Informatics Assoc.*, 19(2): 263–274, 2012. doi: 10.1136/amiajnl-2011-000291. URL https://doi.org/10.1136/amiajnl-2011-000291.

Jin Jing, Emile d'Angremont, Senan Ebrahim, Mohammad Tabaeizadeh, Marcus Ng, Aline Herlopian, Justin Dauwels, and M. Brandon Westover. Rapid annotation of seizures and interictal-ictal-injury continuum eeg patterns. *Journal of Neuroscience Methods*, 347:108956, 2021. ISSN 0165-0270. doi: https://doi.org/10.1016/j.jneumeth.2020.108956. URL https://www.sciencedirect.com/science/article/pii/S0165027020303794.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C. Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. *CoRR*, abs/2108.00106, 2021. URL https://arxiv.org/abs/2108.00106.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. ISRUC-Sleep: A comprehensive public dataset for sleep researchers. *Computer Methods and Programs in Biomedicine*, 2016. ISSN 18727565. doi: 10.1016/j.cmpb.2015.10.013.

J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953. ISSN 00029939, 10886826. URL http://www.jstor.org/stable/2032161.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23528-8.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pp. 12295–12305, 2019.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f8c0c968632845cd133308b1a494967f-Paper.pdf.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2805–2814. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kumar18a.html.

Xingchen Ma and Matthew B. Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7235–7245. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ma21a.html.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15288–15299. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/aeb7b30ef1d024a76f21a1d40e30c302-Paper.pdf.

Allan H. Murphy and Robert L. Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of The Royal Statistical Society Series C-applied Statistics*, 26:41–47, 1977.

Allan H. Murphy and Robert L. Winkler. Probability forecasting in meteorology. *Journal of the American Statistical Association*, 79:489–500, 1984.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102430. URL https://doi.org/10.1145/1102351.1102430.

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

Nicolas Papernot and Patrick D. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *CoRR*, abs/1803.04765, 2018. URL http://arxiv.org/abs/1803.04765.

Kanil Patel, William H. Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=AICNpd8ke-m.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pp. 61–74. MIT Press, 1999.

Zhi Qiao, Austin Bae, Lucas M Glass, Cao Xiao, and Jimeng Sun. FLANNEL (Focal Loss bAsed Neural Network EnsembLe) for COVID-19 detection. *Journal of the American Medical Informatics Association*, 28(3):444–452, 10 2020. ISSN 1527-974X. doi: 10.1093/jamia/ocaa280. URL https://doi.org/10.1093/jamia/ocaa280.

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13456–13467. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/9bc99c590be3511b8d53741684ef574c-Paper.pdf.

Giulio Ruffini, David Ibañez, Marta Castellano, Laura Dubreuil-Vall, Aureli Soria-Frisch, Ron Postuma, Jean-François Gagnon, and Jacques Montplaisir. Deep learning with eeg spectrograms in rapid eye movement behavior disorder. *Frontiers in Neurology*, 10, 2019. ISSN 1664-2295. doi: 10.3389/fneur.2019.00806. URL https://www.frontiersin.org/article/10.3389/fneur.2019.00806.

Danaipat Sodkomkham, Davide Ciliberti, Matthew A. Wilson, Ken ichi Fukui, Koichi Moriyama, Masayuki Numao, and Fabian Kloosterman. Kernel density compression for real-time bayesian encoding/decoding of unsorted hippocampal spikes. *Knowledge-Based Systems*, 94:1–12, 2016. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2015.09.013. URL https://www.sciencedirect.com/science/article/pii/S0950705115003524.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 4278–4284. AAAI Press, 2017.

Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *CoRR*, abs/2105.01601, 2021. URL https://arxiv.org/abs/2105.01601.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3459–3467. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/vaicenavicius19a.html.

Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel). Non-parametric calibration for classification. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 178–190. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/wenger20a.html.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Chaoqi Yang, Zhenbang Wu, Patrick Jiang, Zhen Lin, and Jimeng Sun. PyHealth: A deep learning toolkit for healthcare predictive modeling, 09 2022. URL `https://github.com/sunlabuiuc/PyHealth`.

Ye Yuan, Guangxu Xun, Kebin Jia, and Aidong Zhang. A multi-view deep learning framework for eeg seizure detection. *IEEE Journal of Biomedical and Health Informatics*, 23(1):83–94, 2019. doi: 10.1109/JBHI.2018.2871678.

Bianca Zadrozny and Charles Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 204–213, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113391X. doi: 10.1145/502512.502540. URL `https://doi.org/10.1145/502512.502540`.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pp. 694–699, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775151. URL `https://doi.org/10.1145/775047.775151`.

Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11117–11128. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/zhang20k.html`.

Appendix

Overview of Appendices: Appendix A contains proofs for the lemmata and theorems that appear in Section 3.3. Appendix B clarifies the benefits of the sampling method (equal-size stratified sampling) described in Section 3.4. Appendix C contains more details on the experiments in this paper. Appendix D compares KCal with a simpler variant, namely KCal-Linear, which uses a linear layer as the $\mathbf{\Pi}$. Appendix F explores the effect of d, the projected dimension, on the performance and computational overhead. Finally, Appendix G compares the cross-validation-selected bandwidth vs the analytically computed bandwidth, which shows that it is possible to avoid most of the bandwidth selection steps if we use KCal in an online manner.

A Detailed Assumptions and Proofs

A.1 Assumptions and Definitions

Denote $Z_i := \mathbf{\Pi}(\mathbf{f}(X_i))$ for $i \in [N+1]$. We assume $\{Z_i\}_{i=1}^{N+1}$ are i.i.d. Since fixing $\mathbf{\Pi}$ and $\mathbf{f}$ using $\mathcal{S}_{\text{train}}$, all data will now live in $\mathcal{Z} := \mathbf{\Pi}(\mathbf{f}(\mathcal{X}))$. We are just performing a standard (multivariate) kernel density estimation with only one parameter b on the calibration set. We will use $\hat{g}$ and g to denote the estimation and density in $\mathcal{Z}$, instead of the more cumbersome $\hat{f}_{\mathbf{\Pi}\circ\mathbf{f},k}$ and $f_{\mathbf{\Pi}\circ\mathbf{f},k}$.

Like in Chacón & Duong (2018), we make the following standard assumptions for $g_k(\mathcal{Z})$:

- (For any k) g_k is square integrable and twice differentiable, with all second order partials bounded, continuous and square integrable.

The base "mother kernel" function should satisfy the following (true for the RBF kernel):

- ϕ is spherical symmetric and has a finite second moment. Formally, this means $\int_{\mathbb{R}^d} x\phi(x)dx = \mathbf{0}$ and $\forall i \in [d]$, $\int_{\mathbb{R}^d} x_i x_j \phi(x)dx = \mu_{2,\phi}\mathbf{1}\{i = j\}$ where $\mu_{2,\phi}$ is a fixed finite constant.

In the proof for Lemma 3.1 and Lemma 3.2, for simplicity, we ignore the subscript $_k$ and write g instead of g_k where there is no confusion.

A.2 Proof of Lemma 3.1

Rewriting Eq. (7), we want to show $\|\hat{g}(\mathbf{z}) - g(\mathbf{z})\|_2$ converges to 0 in probability with an admissible $b(m)$, as $m \to \infty$. We first derive the expression of the bias and variance of $\hat{g}$. For the bias, we have:

$$\begin{aligned}
\mathbb{E}[\hat{g}(\mathbf{z})] - g(\mathbf{z}) &= \frac{1}{b^d}\mathbb{E}\left[\phi\left(\frac{\mathbf{z}-Z}{b}\right)\right] - g(\mathbf{z}) && (12)\\
&= \frac{1}{b^d}\int \phi\left(\frac{\mathbf{z}'-\mathbf{z}}{b}\right) g(\mathbf{z}')d\mathbf{z}' - g(\mathbf{z}) && (13)\\
&= \int \phi(\mathbf{u}) g(\mathbf{z}+b\mathbf{u})d\mathbf{u} - g(\mathbf{z}) && (14)\\
&= \int \phi(\mathbf{u})\Big(g(\mathbf{z}) + b(D_g(\mathbf{z}))^\top \mathbf{u} + \frac{1}{2}b^2\mathbf{u}^\top H_g(\mathbf{z})\mathbf{u} + o(\|b\mathbf{u}\|^2)\Big)d\mathbf{u} - g(\mathbf{z}) && (15)\\
&= \int \phi(\mathbf{u})\frac{1}{2}b^2\mathbf{u}^\top H_g(\mathbf{z})\mathbf{u}d\mathbf{u} + o(\|b\mathbf{u}\|^2) && (16)\\
&= \int \phi(\mathbf{u})\frac{1}{2}b^2\sum_{i,j} u_i u_j H_{g,i,j}(\mathbf{z})d\mathbf{u} + o(b^2) && (17)\\
&= \sum_i H_{g,i,i}(\mathbf{z})\mu_{2,\phi}\frac{b^2}{2} + o(b^2) && (18)\\
&= \frac{b^2}{2}\mu_{2,\phi} tr(H_g(\mathbf{z})) + o(b^2) && (19)\\
\implies |\mathbb{E}[\hat{g}(\mathbf{z})] - g(\mathbf{z})| &\le C_{\phi,b} b^2 && (20)
\end{aligned}$$

for some constant $C_{\phi,b}$.

For the variance,

$$\begin{aligned}
Var(\hat{g}(z)) &= Var\left(\frac{1}{mb^d}\sum_{i=1}^m \phi\left(\frac{\mathbf{z}-Z_i}{b}\right)\right) = \frac{1}{mb^{2d}} Var\left(\phi\left(\frac{\mathbf{z}-Z}{b}\right)\right) && (21)\\
&\le \frac{1}{mb^{2d}}\mathbb{E}\left[\phi^2\left(\frac{\mathbf{z}-Z}{b}\right)\right] = \frac{1}{mb^{2d}}\int \phi^2\left(\frac{\mathbf{z}-\mathbf{z}'}{b}\right)g(\mathbf{z}')d\mathbf{z}' && (22)\\
&= \frac{1}{mb^d}\int \phi^2(\mathbf{u})g(\mathbf{z}+b\mathbf{u})d\mathbf{u} && (23)\\
&= \frac{1}{mb^d}\int \phi^2(\mathbf{u})\Big(g(\mathbf{z}) + b(D_g(\mathbf{z}))^\top\mathbf{u} + o(\|b\mathbf{u}\|^1)\Big)d\mathbf{u} && (24)\\
&= \frac{1}{mb^d}\int \phi^2(\mathbf{u})g(\mathbf{z})d\mathbf{u} + o(\frac{1}{mb^d}) && (25)\\
&= \frac{1}{mb^d}g(\mathbf{z})\int \phi^2(\mathbf{u})d\mathbf{u} + o(\frac{1}{mb^d}) \le C_{\phi,v}\frac{1}{mb^d}. && (26)
\end{aligned}$$

for some constant $C_{\phi,v}$.

As a result, for any $\mathbf{z} \in \mathcal{Z}$, we have the MSE as:

$$\begin{aligned}
\mathbb{E}[\|\hat{g}(\mathbf{z}) - g(\mathbf{z})\|^2] &= \mathbb{E}[\|\hat{g}(\mathbf{z}) - \mathbb{E}[\hat{g}(\mathbf{z})] + \mathbb{E}[\hat{g}(\mathbf{z})] - g(\mathbf{z})\|^2] && (27)\\
&= \mathbb{E}[\|\hat{g}(\mathbf{z}) - \mathbb{E}[\hat{g}(\mathbf{z})]\|^2] + \mathbb{E}[\|\mathbb{E}[\hat{g}(\mathbf{z})] - g(\mathbf{z})\|^2] && (28)\\
&= \underbrace{Var(\hat{g}(\mathbf{z}))}_{variance} + \underbrace{(\mathbb{E}[\hat{g}(\mathbf{z})] - g(\mathbf{z}))^2}_{bias^2} && (29)\\
&\le C_{\phi,v}\frac{1}{mb^d} + C_{\phi,b}b^4. && (30)
\end{aligned}$$

This means the MSE goes to 0 as long as $b^d m \to \infty$ and $b \to 0$. As $m \to \infty$, we have $\lim_{m\to\infty} \mathbb{E}[\|\hat{g}(\mathbf{z}) - g(\mathbf{z})\|^2] = 0$.

Now, note that $\hat{g}(\mathbf{z}) = \frac{1}{m}\sum_{j=1}^{m} V_j$ where $V_j = \frac{1}{b^d}\phi(\frac{\mathbf{z}-Z_j}{b})$. By Bernstein's inequality, we have

$$\mathbf{P}\{|\hat{g}(\mathbf{z}) - \mathbb{E}[\hat{g}(\mathbf{z})]| > \epsilon\} \leq 2e^{-\frac{(m\epsilon^2)/2}{mC_{\phi,v}b^{-d}+\frac{1}{3}m\epsilon\phi(0)b^{-d}}} \leq e^{-Bmb^d\epsilon^2} \tag{31}$$

for some constant B as long as ϵ is smaller than a constant (say 1). With triangular inequality, we have

$$\mathbf{P}\{|\hat{g}(\mathbf{z}) - g(\mathbf{z})| > \epsilon + C_{\phi,b}b^2\} \leq \mathbf{P}\{|\hat{g}(\mathbf{z}) - \mathbb{E}[\hat{g}(\mathbf{z})]| > \epsilon\} \leq e^{-Bmb^d\epsilon^2} \tag{32}$$

which gives us the conclusion as the RHS goes to 0 as $m \to \infty$.

A.3 Proof of Lemma 3.2

Lemma 3.2 says that $b = \Theta(m^{-\frac{1}{d+4}})$ is the optimal shrinkage rate to minimize $\mathbb{E}[\|\hat{g}(\mathbf{z}) - g(\mathbf{z})\|^2]$. Following Eq. (30), by letting $C_{\phi,b}\frac{1}{mb^d} = C_{\phi,v}b^4$, we get $b = \Theta(m^{-\frac{1}{d+4}})$. We can also derive this formula by taking the derivative of Eq. (30) with respect to b and setting it to 0, which gives us (asymptotically):

$$\frac{-dC_{\phi,b}}{m}b^{-(d+1)} + 4C_{\phi,v}b^3 = 0 \Rightarrow b^* = C'm^{-\frac{1}{d+4}} \tag{33}$$

for some constant C'. The optimal MSE is thus $O(m^{-\frac{4}{d+4}})$.

A.4 Proof of Theorem 3.3

Denote $m := \min_k\{m_k\}$. $\forall k \in [K]$, Bernstein's inequality[6] gives us:

$$\mathbb{P}\{|\hat{\pi}_k - \pi_k| \geq \epsilon_1\} \leq 2e^{-\frac{N\epsilon_1^2}{2v_{min}+\frac{2}{3}\epsilon_1}} \leq e^{-B_2N\epsilon_1^2} \tag{35}$$

where $v_{min} = \min_k\{\pi_k(1-\pi_k)\}$ and some constant B_2 (we find the smallest such constant among all classes), as long as ϵ_1 is smaller than a constant (e.g. 1).

From Eq. 32, with $b = C'm^{\frac{-1}{d+4}}$, and let $\epsilon = m^{\frac{-\lambda}{d+4}}$ for $\lambda \in (0, 2)$, we have, for some constants C_1, B_1:

$$\mathbb{P}\{|\hat{g}(\mathbf{z}) - g(\mathbf{z})| > C_1 m^{\frac{-\lambda}{d+4}}\} \leq e^{-B_1 m^{\frac{4-2\lambda}{d+4}}}. \tag{36}$$

Define $\delta_k := e^{-B_1 m_k^{\frac{4-2\lambda}{d+4}}} + e^{-B_2N\epsilon_1^2} \leq e^{-B_1 m^{\frac{4-2\lambda}{d+4}}} + e^{-B_2N\epsilon_1^2}$. With probability $\geq 1 - \sum_k \delta_k$ (union bound), for all k:

$$\begin{aligned} |\hat{g}_k(\mathbf{z})\hat{\pi}_k - g_k(\mathbf{z})\pi_k| &\leq |\hat{g}_k(\mathbf{z})\hat{\pi}_k - g_k(\mathbf{z})\hat{\pi}_k| + |g_k(\mathbf{z})\hat{\pi}_k - g_k(\mathbf{z})\pi_k| && (37)\\ &\leq C_1 m_k^{\frac{-\lambda}{d+4}} + g_k(\mathbf{z})\epsilon_1. && (38) \end{aligned}$$

[6] One could apply Bennett's inequality to get:

$$\mathbb{P}\{|\hat{\pi}_k - \pi_k| \geq \epsilon_1\} \leq e^{-N\frac{\pi_k}{1-\pi_k}h(\frac{\epsilon_1}{\pi_k})} + e^{-N\frac{1-\pi_k}{\pi_k}h(\frac{\epsilon_1}{1-\pi_k})} \tag{34}$$

and repeat the following proof for a slightly tigher bound. However, the notation is much more complicated.

Denote $g^{+}(\mathbf{z}) = \max_k g_k(\mathbf{z}), g^{-}(\mathbf{z}) = \min_k g_k(\mathbf{z})$, and $\overline{g}(\mathbf{z}) = \sum_k g_k(\mathbf{z})\pi_k$. Denote $\epsilon_{k,2} := C_1 m_k^{\frac{-\lambda}{d+4}}$, Eq. 38 means:

$$\hat{p}_k(\mathbf{z}) = \frac{\hat{g}_k(\mathbf{z})\hat{\pi}_k}{\sum_{k'} \hat{g}_{k'}(\mathbf{z})\hat{\pi}_{k'}} \geq \frac{g_k(\mathbf{z})\pi_k - \epsilon_{k,2} - g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z}) + \sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]} \tag{39}$$

$$= \frac{g_k(\mathbf{z})\pi_k - \epsilon_{k,2} - g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})} \frac{1}{1 + \frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})}} \tag{40}$$

$$\geq \frac{g_k(\mathbf{z})\pi_k - \epsilon_{k,2} - g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})}(1 - \frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})}) \tag{41}$$

$$\geq p_k(\mathbf{z}) - \frac{\epsilon_{k,2} + g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})} - \frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})} \tag{42}$$

Similarly,

$$\hat{p}_k(\mathbf{z}) \leq \frac{g_k(\mathbf{z})\pi_k + \epsilon_{k,2} + g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})} \frac{1}{1 - \frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})}} \tag{43}$$

$$\leq (p_k(\mathbf{z}) + \frac{\epsilon_{k,2} + g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})})(1 + 2\frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})}) \tag{44}$$

$$\leq p_k(\mathbf{z}) + \frac{\epsilon_{k,2} + g_k(\mathbf{z})\epsilon_1}{\overline{g}(\mathbf{z})} + \frac{3\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})} \tag{45}$$

We can proceed from Eq.43 to 44 and from 44 to 45 when $\frac{\sum_{k'}[\epsilon_{k',2} + g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})} \leq 0.5$, which is achievable for a large m (the smallest m_k, thus N) given any $\mathbf{z}$ as long as $\overline{g}(\mathbf{z}) > 0$. .

With Eq. 42 and 45, with probability $\geq 1 - K(e^{-B_1 m^{\frac{4-2\lambda}{d+4}}} + e^{-B_2 N \epsilon_1^2})$:

$$|p_k(\mathbf{z}) - \hat{p}_k(\mathbf{z})| \leq \frac{(3K+1)(\epsilon_{k,2} + g^{+}(\mathbf{z})\epsilon_1)}{\overline{g}(\mathbf{z})} \tag{46}$$

$$= \frac{(3K+1)(C_1 m^{\frac{-\lambda}{d+4}} + g^{+}(\mathbf{z})\epsilon_1)}{\overline{g}(\mathbf{z})} \tag{47}$$

If we let $\epsilon_1 = \Theta(m^{\frac{-\lambda}{d+4}})$ and merge the constants, we have, with probability $\geq 1 - Ke^{-Bm^{\frac{4-2\lambda}{d+4}}}$ (note that $N \geq Km$):

$$|p_k(\mathbf{z}) - \hat{p}_k(\mathbf{z})| \leq (3K+1)Cm^{\frac{-\lambda}{d+4}} \tag{48}$$

$$\implies |\mathbf{p}(\mathbf{z}) - \hat{\mathbf{p}}(\mathbf{z})|_1 \leq K(3K+1)Cm^{\frac{-\lambda}{d+4}} \tag{49}$$

with some constant C and B that depends on $\{g_k(\mathbf{z})\}_{k \in [K]}$.

A.5 Proof of Theorem 3.4

If we assume g_k is α-Hölder continuous for all k then by Theorem 2 in Jiang (2017), there exists positive constant C' independent of b and m, such that the following holds with probability $\geq 1 - \frac{1}{m_k}$

$$\sup_{\mathbf{z}} |\hat{g}_k(\mathbf{z}) - g_k(\mathbf{z})| < C'\Big(b^{\alpha} + \sqrt{\frac{\log m_k}{m_k b^d}}\Big). \tag{50}$$

Furthermore, we assume that all the densities are bounded from below (see, for example, Section 3 in Gadat et al. (2016)). Denote $U := \max_k \sup_{\mathbf{z}} g_k(\mathbf{z})$ and $L := \min_k \inf_{\mathbf{z}} g_k(\mathbf{z})$.

We could replace $\epsilon_{k,2}$ in the previous section with $\epsilon_{k,2} = C_1(b^{\alpha} + \sqrt{\frac{\log m_k}{m_k b^d}})$. Following similar steps leading towards Eq. 42 and Eq. 45, we have, with probability $\geq 1 - K(\frac{1}{m} + e^{-B_2 N \epsilon_1^2})$, for any $\mathbf{z}$:

$$|\hat{p}_k(\mathbf{z}) - p_k(\mathbf{z})| \leq \frac{(3K+1)(\epsilon_{k,2} + g^{+}(\mathbf{z})\epsilon_1)}{\overline{g}(\mathbf{z})} \tag{51}$$

$$\leq \frac{(3K+1)}{L}\Big(C_1(b^{\alpha} + \sqrt{\frac{\log m}{mb^d}}) + U\epsilon_1\Big) \tag{52}$$

Note that we still need $\frac{\sum_{k'}[\epsilon_{k',2}+g_{k'}(\mathbf{z})\epsilon_1]}{\overline{g}(\mathbf{z})} \leq 0.5$, which is satisfied as N increases because $g_k(\mathbf{z}) >= L$. Now, we let $b = \Theta((\frac{\log m}{m})^{\frac{1}{d+2\alpha}})$, and $\epsilon_1 = \Theta((\frac{\log m}{m})^{\frac{\alpha}{d+2\alpha}})$, we have with probability $\geq 1 - K(m^{-1} + e^{-Bm^{\frac{d}{d+2\alpha}}(\log m)^{\frac{2\alpha}{d+2\alpha}}}) = 1 - K(m^{-1} + m^{-B\frac{2\alpha}{d+2\alpha}m^{\frac{d}{d+2\alpha}}})$:

$$|\hat{p}_k(\mathbf{z}) - p_k(\mathbf{z})| \leq (3K+1)C(\frac{\log m}{m})^{\frac{\alpha}{d+2\alpha}}. \tag{53}$$

Finally, with probability $\geq 1 - K(m^{-1} + m^{-B\frac{2\alpha}{d+2\alpha}m^{\frac{d}{d+2\alpha}}})$, for any $\mathbf{q}$ in Δ^{K-1}:

$$\sup_{\mathbf{z}:\hat{\mathbf{p}}(\mathbf{z})=\mathbf{q}} |\mathbb{P}\{Y=k|\hat{\mathbf{p}}(\mathbf{z})=\mathbf{q}\} - q_k| = \sup_{\mathbf{z}:\hat{\mathbf{p}}(\mathbf{z})=\mathbf{q}} |p_k(\mathbf{z}) - \hat{p}_k(\mathbf{z})| \leq \sup_{\mathbf{z}} |p_k(\mathbf{z}) - \hat{p}_k(\mathbf{z})| \leq (3K+1)C(\frac{\log m}{m})^{\frac{\alpha}{d+2\alpha}}. \tag{54}$$

B Theoretical Analysis of Equal-sized Stratified Sampling in Training

We adopted equal-sized stratified sampling to facilitate efficient training. Here we provide some theoretical justification of this choice.

After fixing a x_0 whose label y_0 is the prediction target, the problem is essentially estimating $\frac{\mu_k p_k}{\sum_{k'} \mu_{k'} p_{k'}}$ for all k, where p_k denotes the frequency of class k in the population[7] and μ_k denotes $\mathbb{E}[\phi(X, x_0)|Y=k]$. Note that we know p_k, but not μ_k, since p_k is fixed for our training set, but μ_k depends on x_0 and $\mathbf{\Pi}$, which is what we are training. Suppose we can afford to use M samples in total to make the prediction, the question is: How do we distribute these M samples to different classes?

What sampling method to use will depend on many factors, although a stratified sampling strategy tends to be more efficient in sample size. The sampling method we use (sample the same number of samples for each class k) intuitively will improve the estimation quality of the rarer class. Here, we will elaborate why we chose this sampling method, the assumptions behind it, and why it helps training.

Denoting $S_k = \mu_k p_k$ and $S_{-k} = \sum_{k' \neq k} \mu_{k'} p_{k'}$, we can apply Taylor expansion to get an approximation of the variance[8]:

$$Var(\frac{S_k}{S_{-k}+S_k}) \approx \frac{1}{\mathbb{E}[S_{-k}+S_k]^2} Var(S_k) - 2\frac{\mathbb{E}[S_k]}{\mathbb{E}[S_{-k}+S_k]^3} Cov(S_k, S_{-k}+S_k) \tag{55}$$

$$+ \frac{\mathbb{E}[S_k]^2}{\mathbb{E}[S_{-k}+S_k]^4} Var(S_{-k}+S_k) \tag{56}$$

If we perform stratified sampling of any kind, then $Cov(S_k, S_{-k}) = 0$, and Eq. (56) becomes:

$$Var(\frac{S_k}{S_{-k}+S_k}) \approx \frac{1}{\mathbb{E}[S_{-k}+S_k]^2} Var(S_k) - 2\frac{\mathbb{E}[S_k]}{\mathbb{E}[S_{-k}+S_k]^3} Var(S_k) \tag{57}$$

$$+ \frac{\mathbb{E}[S_k]^2}{\mathbb{E}[S_{-k}+S_k]^4} [Var(S_{-k}) + Var(S_k)] \tag{58}$$

$$= \frac{\mathbb{E}[S_k]^2}{\mathbb{E}[S_{-k}+S_k]^4} \left(\left[\frac{\mathbb{E}[S_{-k}]}{\mathbb{E}[S_k]} \right]^2 Var(S_k) + Var(S_{-k}) \right) \tag{59}$$

To further analyze Eq. (59) and gain more intuition, we make the following assumptions:

- For any $k' \neq y_0$, $\mu_{k'}$ has the same value denoted as μ_{-y_0} (and is smaller than μ_{y_0}). Intuitively, this is like considering a one-vs-rest classification problem, and we are just saying data from the same class will look more similar according to our kernel.

[7] In our case, this population is the large training set.

[8] Such a derivation could be found in https://www.stat.cmu.edu/~hseltman/files/ratio.pdf

- The standard deviation for a single observation is directly proportional to the mean. Namely, for all k, $\frac{\sqrt{Var(\phi(X, x_0)|Y = k)}}{\mathbb{E}[\phi(X, x_0)|Y = k]} \equiv r$ for a fixed number r.

If we assign m_k samples to estimate μ_k then we have $Var(S_k) = r^2 \frac{\mathbb{E}[S_k]^2}{m_k}$ and $Var(S_{-k}) = r^2 \frac{\mathbb{E}[S_{-k}]^2}{M - m_k}$, where $M = \sum_{k'=1}^{K} m_{k'}$ ($M \gg m_k$ when K is large). This transforms Eq. (59) into:

$$Var\left(\frac{S_k}{S_{-k} + S_k}\right) \approx \frac{\mathbb{E}[S_k]^2 \mathbb{E}[S_{-k}]^2}{\mathbb{E}[S_{-k} + S_k]^4} r^2 \left(\frac{1}{m_k} + \frac{1}{M - m_k}\right) = C\left(\frac{1}{m_k} + \frac{1}{M - m_k}\right) \tag{60}$$

where C is a constant that does not depend on m_k.

Without prior information, it is natural to assume C is class-independent (or at least relatively constant across classes). Now, if our goal is to minimize the average variance, by Cauchy-Schwarts inequality we have:

$$\sum_{k=1}^{K} \frac{1}{m_k} \geq \frac{K^2}{M} \tag{61}$$

$$\sum_{k=1}^{K} \frac{1}{M - m_k} \geq \frac{K^2}{(K-1)M} \tag{62}$$

The equality in both cases is achieved if and only if $m_k \equiv \frac{M}{K}$ for all k. This means, to minimize the average variance $\frac{C}{K} \sum_{k=1}^{K} (\frac{1}{m_k} + \frac{1}{M - m_k})$, we need to choose m_k to be the same for all class k.

It is worth noting that the discussion above is about training (and how to get better estimation therein). This is not referring to errors of the final $\mathbf{\Pi}$. Given enough time, different ways to sample data lead to similar performance.

C Additional Experimental Details

C.1 Datasets

This section provides more detail on the healthcare datasets, which might be less familiar to readers.

IIIC (Jing et al., 2021; Ge et al., 2021) is an electroencephalography (EEG) dataset from the Massachusetts General Hospital EEG Archive. It is collected for the purpose of automated ictal-interictal-injury-continuum (IIIC) detection/monitoring. IIIC patterns include seizure and seizure-like patterns designated Lateralized Periodic Discharges (LPDs), Generalized Periodic Discharges (GPDs), Lateralized Rhythmic Delta Activity (LRDA), and Generalized Rhythmic Delta Activity (GRDA)(Ge et al., 2021). The training data has been enriched with "label spreading" (Ge et al., 2021), whereas the test (and calibration) data consists of only labels from medical experts. To improve stability (because IIIC labeling is a challenging task for even experts), any sample with less than 3 labels are dropped. The majority label is then used as the truth for the test and calibration ses. For more details on how the data was collected and labeled, please refer to Jing et al. (2021); Ge et al. (2021).

ISRUC (Khalighi et al., 2016) is a public polysomnographic (PSG) dataset for the sleep staging task. It has three groups of data, with the first group having the most data and most widely used. The (group 1) dataset contains 100 subjects with one recording session per subject. Every 30 second of the recording is considered an "epoch" and is rated independently by two human experts. We use the label from the first expert as the gold label. The five classes of ISRUC correspond to five different stages of sleep, including Rapid Eye Movement (REM), Non-REM Stage 1 (N1), Non-REM Stage 2 (N2), Non-REM Stage 3 (N3), and Wake (Wake). For more details, please refer to Khalighi et al. (2016).

PhysioNet Callenge 2017 (PN2017) (Clifford et al., 2017; Goldberger et al., 2000) is a public (upon request) electrocardiogram (ECG) dataset for ECG rhythm classification. The ECG recordings are sampled at 300Hz. The original dataset contains four classes: Normal sinus rhythm (N), Atrial Fibrillation (AF), Other cardiac rhythms (O) and Noise segment. Among these patterns, AF is an abnormal heart rhythm, and is the "important class". We used the same processing method as Hong et al. (2019), which cuts one segment into several shorter segments with data augmentation during the training phase.

A summary of the classes can be found below in Table 7.

Table 7: Additional information about the healthcare datasets used in this paper.

	IIIC			ISRUC			PN2017		
Dataset	Name	Train	Cal+Test	Name	Train	Cal+Test	Name	Train	Cal+Test
Class 0	Other	42228	6852	Wake	14325	6433	Normal	8877	2893
Class 1	Seizure	3305	549	N1	7589	3798	Other	4524	1579
Class 2	LPD	17338	7589	N2	19501	8505	AF	1345	449
Class 3	GPD	16983	9737	N3	12012	5254	Noisy	341	145
Class 4	LRDA	12515	5946	REM	8414	3452	–	–	–
Class 5	GRDA	11449	5067	–	–	–	–	–	–

Data Licenses and Consent:

- ISRUC: We could not find the license. Per Khalighi et al. (2016), "All patients referred were submitted to an initial briefing with the support of an informed consent document. The ethics committee of CHUC approved the use of the data of the referred patients as anonymous for the research purposes".
- PN2017: The license is Open Data Commons Attribution License v1.0. The dataset is donated by AliveCor.
- IIIC: We could not find the license. Per Jing et al. (2021) "the local IRB waived the requirement for informed consent for this retrospective analysis of EEG data".
- CIFAR-100/CIFAR-10: We could not find the license. They are publicly available.
- SVHN: Under CC0: Public Domain license. It is publicly available.

C.2 Baseline Implementation

- Temperature Scaling: We used the github repository accompanying Guo et al. (2017), `https://github.com/gpleiss/temperature_scaling`.
- Dirichlet Calibration: We used the code at `https://github.com/dirichletcal/experiments_dnn`.
- Focal Loss (Mukhoti et al., 2020): We used the loss function and the gamma schedule provided in `https://github.com/torrvision/focal_calibration`, and replaced our CrossEntropy loss function in all experiments during training.
- Mutual-information-maximization-based Binning (`I-Max`): We use the official github implementation `https://github.com/boschresearch/imax-calibration`. To normalize and get valid probability vectors, we used softmax on the log-odds given by `I-Max`.
- Gaussian Process Calibration: We use the official github implementation `https://github.com/JonathanWenger/pycalib`.
- Splines-based Calibration: We use the official github implementation `https://github.com/kartikgupta-at-anu/spline-calibration`.
- Intra Order-preserving Calibration: We use the official github implementation `https://github.com/AmirooR/IntraOrderPreservingCalibration`.
- MMCE: We use the official github implementation `https://github.com/aviralkumar2907/MMCE` with additional temperature scaling on the calibration set as suggested in the original paper.

C.3 Training Details

For CIFAR-10, CIFAR-100, SVHN, and ISRUC, the models are trained for 50 epochs, using a one-cycle Cosine scheduler with 3 warm-up and 10 cool-down epochs (the other parameters are default in `timm`). The exact ViT and Mixer are `vit_base_patch16_224_in21k` and `mixer_b16_224_in21k` implemented and pretrained by `timm` . For PN2017, the number of epochs is 100, and we use a `ReduceLROnPlateau` scheduler that halves the learning rate with the patience parameter set to 10 epochs. We use a batch size of 128, SGD optimizer and weight decay rate of 1e-4. For IIIC dataset, we use a AdamW optimizer with a weight decay rate of 1e-5, and no scheduler. The learning rates are 2e-4 for CIFAR-10, CIFAR-100 and SVHN, 5e-3 for ISRUC, 1e-2 for PN2017 and 1e-3 for IIIC. For all datasets except for IIIC, we used `LabelSmoothingCrossEntropy` in `timm` with smoothing being 0.1. For IIIC, since the original dataset contains pseudo-labels that form a distribution, we use a cross entropy loss. The experiments for the `Focal` baseline replace all loss functions with the proposed focal loss.

To train $\mathbf{\Pi}$, we use an SGD optimizer with a learning rate of 4e-4 for CIFAR-10, CIFAR-100, SVHN and IIIC, 1e-3 for ISRUC and PN2017. We use `ReduceLROnPlateau` scheduler that halves the learning rate with the patience parameter set to 10 epochs, and trains for 100 epochs. Each epoch has a fixed number of 5000 batches (regardless of the size of the training set) and each batch consists of $B = 64$ prediction samples and a "background" set used to construct KDE with $m_k \equiv m = 20$ for all k. The exact details could be found in our code. Training time for the largest dataset (except for ImageNet), SVHN, is 3 hours for the base neural network, and 1 hour for $\mathbf{\Pi}$ on a machine with Nvidia RTX 3090 GPU. Inference time is much shorter.

C.4 Additional Evaluation Metrics

In this section, we compute the following variants of the evaluation metrics presented in the main text. The conclusion stays very similar across all methods.

- The static (equal-width bins) version of CECE, in Table 8.
- The static (equal-width bins) version of ECE, in Table 9.
- The multi-class version of Brier score, in Table 10. To be specific, the brier score in the main text is $\frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{p}}_{k_i^*}(x_i) - \mathbb{1}\{y_i = k_i^*\})^2$ where $k_i^* = \arg\max_k \hat{\mathbf{p}}_k(x_i)$. The multi-class version of Brier score is $\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}(\hat{\mathbf{p}}_k(x_i) - \mathbb{1}\{y_i = k\})^2$.
- NLL Loss, in Table 11.

Table 8: (Static) Class-wise ECE in 10^{-2} (↓ means lower=better). The best accuracy-preserving method is in **bold** (p=0.01). The otherwise lowest number is underscored. KCal almost always achieves the lowest class-wise ECE, while maintaining accuracy.

CECE ↓	UnCal	TS	DirCal	I-Max	Focal	Spline	IOP	GP	MMCE	KCal
IIIC (pat)	8.01±0.27	8.94±0.86	**5.11±1.49**	9.17±0.99	8.95±0.52	8.55±0.63	8.30±0.53	7.94±0.65	7.09±0.44	**4.66±1.30**
IIIC	7.89±0.02	8.96±0.50	**2.13±0.13**	8.77±0.24	8.76±0.02	8.41±0.23	7.97±0.26	7.51±0.24	6.66±0.26	**2.04±0.27**
ISRUC (pat)	**4.51±0.25**	**4.68±0.77**	**4.19±0.89**	8.65±0.99	9.24±0.20	**4.67±0.46**	**4.59±0.59**	**4.63±0.42**	**4.06±0.35**	**3.84±1.22**
ISRUC	4.53±0.02	5.18±0.79	2.73±0.38	9.29±0.86	9.07±0.02	4.75±0.16	4.69±0.37	4.71±0.25	4.07±0.21	**1.93±0.27**
PN2017	12.20±0.07	12.32±0.19	**4.04±0.54**	9.70±1.19	16.70±0.10	8.42±0.73	12.10±0.37	12.20±0.07	12.20±0.32	**3.83±1.27**
C10 (ViT)	3.42±0.01	1.39±0.08	1.25±0.08	**1.15±0.06**	5.19±0.03	1.36±0.06	1.25±0.07	1.23±0.06	1.52±0.22	**1.18±0.08**
C10 (Mixer)	3.36±0.02	2.11±0.11	1.64±0.08	**1.76±0.24**	7.02±0.03	1.71±0.09	1.78±0.10	1.75±0.10	1.95±0.27	**1.59±0.06**
C100 (ViT)	6.33±0.05	6.43±0.29	5.44±0.14	5.96±0.21	6.07±0.05	**5.16±0.17**	5.58±0.14	5.54±0.09	5.30±0.22	**5.06±0.11**
C100 (Mixer)	5.60±0.05	6.75±0.25	5.87±0.20	6.64±0.29	6.08±0.06	5.56±0.13	6.09±0.32	5.80±0.14	6.15±0.21	**5.16±0.07**
SVHN (ViT)	3.50±0.01	2.56±0.58	**1.40±0.06**	2.98±0.22	6.11±0.02	1.51±0.07	**1.47±0.07**	1.51±0.05	1.63±0.11	**1.46±0.08**
SVHN (Mixer)	3.36±0.02	3.38±0.67	**1.39±0.11**	3.00±0.16	5.79±0.02	1.66±0.07	1.54±0.06	1.58±0.06	1.73±0.09	1.57±0.11
ImageNet	3.70±0.03	3.99±0.07	6.11±0.22	3.29±0.21	–	**2.80±0.07**	**2.93±0.16**	3.05±0.08	–	2.40±0.04

Table 9: (Static) ECE in 10^{-2} (↓ means lower=better). The best accuracy-preserving method is in **bold** (p=0.01). KCal is usually on par or better than the best baseline.

ECE ↓	UnCal	TS	DirCal	I-Max	Focal	Spline	IOP	GP	MMCE	KCal
IIIC (pat)	9.18±1.08	**4.95±2.77**	**2.87±1.62**	10.56±4.05	7.37±0.53	**4.54±2.07**	**4.56±2.15**	**3.84±1.63**	6.34±3.30	**4.28±1.42**
IIIC	9.13±0.04	4.42±1.53	**1.22±0.17**	10.17±0.81	7.10±0.04	3.08±0.65	3.44±0.38	1.68±0.55	4.78±2.26	2.55±0.61
ISRUC (pat)	3.60±0.32	**2.70±1.56**	2.91±1.02	8.82±1.41	14.95±0.40	**1.99±0.36**	**2.40±1.43**	**1.94±0.62**	**2.09±0.97**	**2.74±1.29**
ISRUC	3.46±0.06	3.81±1.67	2.20±0.68	9.58±1.26	14.76±0.05	**1.48±0.55**	2.69±0.94	2.04±0.76	**2.08±1.06**	**1.34±0.41**
PN2017	17.10±0.14	17.34±0.42	**5.46±0.66**	8.97±1.85	24.65±0.13	**6.10±2.22**	16.55±2.03	17.13±0.15	13.21±1.08	**4.56±1.41**
C10 (ViT)	9.17±0.05	0.76±0.11	0.44±0.08	0.61±0.06	7.19±0.06	0.49±0.10	0.38±0.05	**0.28±0.07**	0.65±0.15	0.41±0.10
C10 (Mixer)	9.06±0.05	1.11±0.12	0.51±0.05	1.04±0.17	12.54±0.06	0.48±0.08	0.56±0.12	**0.34±0.06**	1.01±0.40	0.65±0.09
C100 (ViT)	11.65±0.14	2.81±0.44	**0.77±0.12**	3.39±0.23	9.98±0.09	1.07±0.24	1.24±0.27	0.92±0.12	1.21±0.36	1.58±0.33
C100 (Mixer)	13.71±0.15	3.18±0.35	1.17±0.26	4.82±0.25	14.36±0.20	**1.20±0.35**	1.82±0.72	**1.15±0.22**	2.14±0.49	3.11±0.48
SVHN (ViT)	10.11±0.05	**2.44±2.72**	**0.61±0.09**	2.08±0.18	12.17±0.08	**0.64±0.14**	**0.55±0.11**	**0.61±0.10**	**0.66±0.15**	0.71±0.13
SVHN (Mixer)	10.30±0.04	3.19±2.55	**0.57±0.08**	2.21±0.10	11.09±0.06	0.67±0.13	**0.49±0.10**	0.62±0.08	0.69±0.21	0.74±0.11
ImageNet	3.06±0.13	3.26±0.13	4.26±0.74	8.05±0.32	–	1.13±0.15	1.38±0.46	**0.95±0.16**	–	1.30±0.28

Table 10: Brier Score (multi-class) in 10^{-2} (↓ means lower=better). The best accuracy-preserving methods are in **bold** (p=0.01).

Brier ↓	UnCal	TS	DirCal	I-Max	Focal	Spline	IOP	GP	MMCE	KCal
IIIC (pat)	9.23±0.18	9.11±0.31	**8.13±0.26**	9.22±0.44	9.69±0.20	9.05±0.25	9.07±0.27	9.01±0.24	9.13±0.25	**8.38±0.36**
IIIC	9.25±0.01	9.10±0.06	7.86±0.02	9.17±0.05	9.68±0.00	9.00±0.01	9.05±0.03	8.95±0.03	9.07±0.06	**7.40±0.04**
ISRUC (pat)	6.84±0.17	6.83±0.17	6.83±0.21	7.15±0.21	7.97±0.14	6.79±0.17	6.82±0.18	6.80±0.17	**6.59±0.14**	**6.67±0.18**
ISRUC	6.95±0.02	6.97±0.06	6.66±0.04	7.31±0.11	8.07±0.01	6.90±0.02	6.94±0.04	6.90±0.03	6.68±0.02	**6.30±0.03**
PN2017	14.92±0.02	14.97±0.11	**12.85±0.09**	14.03±0.20	17.64±0.01	13.78±0.13	14.84±0.24	14.92±0.02	15.26±0.17	**12.81±0.13**
C10 (ViT)	0.27±0.01	0.18±0.01	**0.16±0.01**	0.17±0.01	0.31±0.01	**0.16±0.01**	**0.16±0.01**	**0.16±0.01**	0.18±0.01	**0.15±0.01**
C10 (Mixer)	0.39±0.01	0.30±0.01	0.30±0.01	0.30±0.02	0.74±0.01	**0.29±0.01**	**0.29±0.01**	**0.28±0.01**	**0.30±0.03**	**0.28±0.01**
C100 (ViT)	0.14±0.00	0.12±0.00	0.12±0.00	0.12±0.00	0.14±0.00	0.12±0.00	0.12±0.00	0.12±0.00	**0.11±0.00**	0.11±0.00
C100 (Mixer)	0.21±0.00	0.19±0.00	0.18±0.00	0.19±0.00	0.23±0.00	0.18±0.00	0.18±0.00	0.18±0.00	**0.17±0.00**	0.18±0.00
SVHN (ViT)	0.76±0.01	0.65±0.04	0.62±0.01	0.65±0.01	0.95±0.01	0.62±0.01	0.62±0.01	0.62±0.01	**0.54±0.00**	0.55±0.01
SVHN (Mixer)	0.77±0.01	0.68±0.05	0.62±0.01	0.66±0.01	0.97±0.01	0.63±0.01	0.63±0.01	0.63±0.01	0.68±0.01	**0.60±0.01**
ImageNet	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

Table 11: NLL (↓ means lower=better). The best accuracy-preserving methods are in **bold** (p=0.01).

NLL ↓	UnCal	TS	DirCal	I-Max	Focal	Spline	IOP	GP	MMCE	KCal
IIIC (pat)	1.09±0.02	1.08±0.05	**0.97±0.05**	1.11±0.07	1.11±0.03	1.07±0.03	1.07±0.04	1.06±0.03	1.07±0.03	**1.00±0.05**
IIIC	1.09±0.00	1.08±0.01	0.92±0.00	1.10±0.01	1.11±0.00	1.06±0.00	1.06±0.00	1.05±0.00	1.06±0.01	**0.87±0.01**
ISRUC (pat)	0.63±0.02	0.62±0.02	**0.62±0.02**	0.69±0.03	0.72±0.01	**0.61±0.02**	0.62±0.02	**0.61±0.02**	**0.60±0.02**	**0.61±0.02**
ISRUC	0.64±0.00	0.63±0.01	0.60±0.00	0.71±0.02	0.73±0.00	0.63±0.00	0.63±0.00	0.62±0.00	0.61±0.00	**0.57±0.00**
PN2017	1.00±0.00	1.00±0.00	**0.86±0.01**	0.96±0.02	1.19±0.00	0.95±0.01	0.99±0.02	1.00±0.00	1.04±0.03	**0.86±0.01**
C10 (ViT)	0.12±0.00	0.05±0.01	**0.03±0.00**	0.04±0.00	0.10±0.00	0.04±0.00	**0.03±0.00**	**0.03±0.00**	0.05±0.00	**0.03±0.00**
C10 (Mixer)	0.15±0.00	0.07±0.01	0.06±0.00	0.07±0.00	0.20±0.00	**0.06±0.00**	0.06±0.00	**0.06±0.00**	**0.07±0.02**	**0.06±0.00**
C100 (ViT)	0.38±0.00	0.29±0.01	0.28±0.01	0.32±0.01	0.36±0.00	0.28±0.01	0.28±0.01	0.27±0.01	**0.25±0.01**	0.27±0.00
C100 (Mixer)	0.54±0.01	0.43±0.01	0.43±0.01	0.47±0.01	0.54±0.01	0.43±0.01	0.43±0.01	0.42±0.01	**0.39±0.01**	0.44±0.01
SVHN (ViT)	0.23±0.00	0.16±0.02	0.15±0.00	0.17±0.00	0.26±0.00	0.15±0.00	0.15±0.00	0.15±0.00	**0.13±0.00**	0.13±0.00
SVHN (Mixer)	0.23±0.00	0.19±0.03	0.15±0.00	0.18±0.00	0.26±0.00	0.16±0.00	0.15±0.00	0.15±0.00	0.16±0.00	**0.15±0.00**
ImageNet	0.84±0.01	0.83±0.01	0.90±0.02	0.87±0.02	–	0.77±0.01	**0.75±0.01**	**0.75±0.01**	–	0.87±0.01

C.5 Reliability Diagrams

Figure 3, 4, and 5 are the reliability diagrams for the IIIC, ISRUC and PN2017 dataset, respectively. We keep only bins with at least 15 samples, because otherwise the "gap" is misleading due to small sample and big variance. The count of samples in each bin is plotted on the right axis (log-scale). The conclusion is similar. In all cases, TS seems to calibrate the overall ECE but fails on some classes. DirCal tends to improve on all classes, but KCal usually closes the gap between actual frequency and the prediction further.

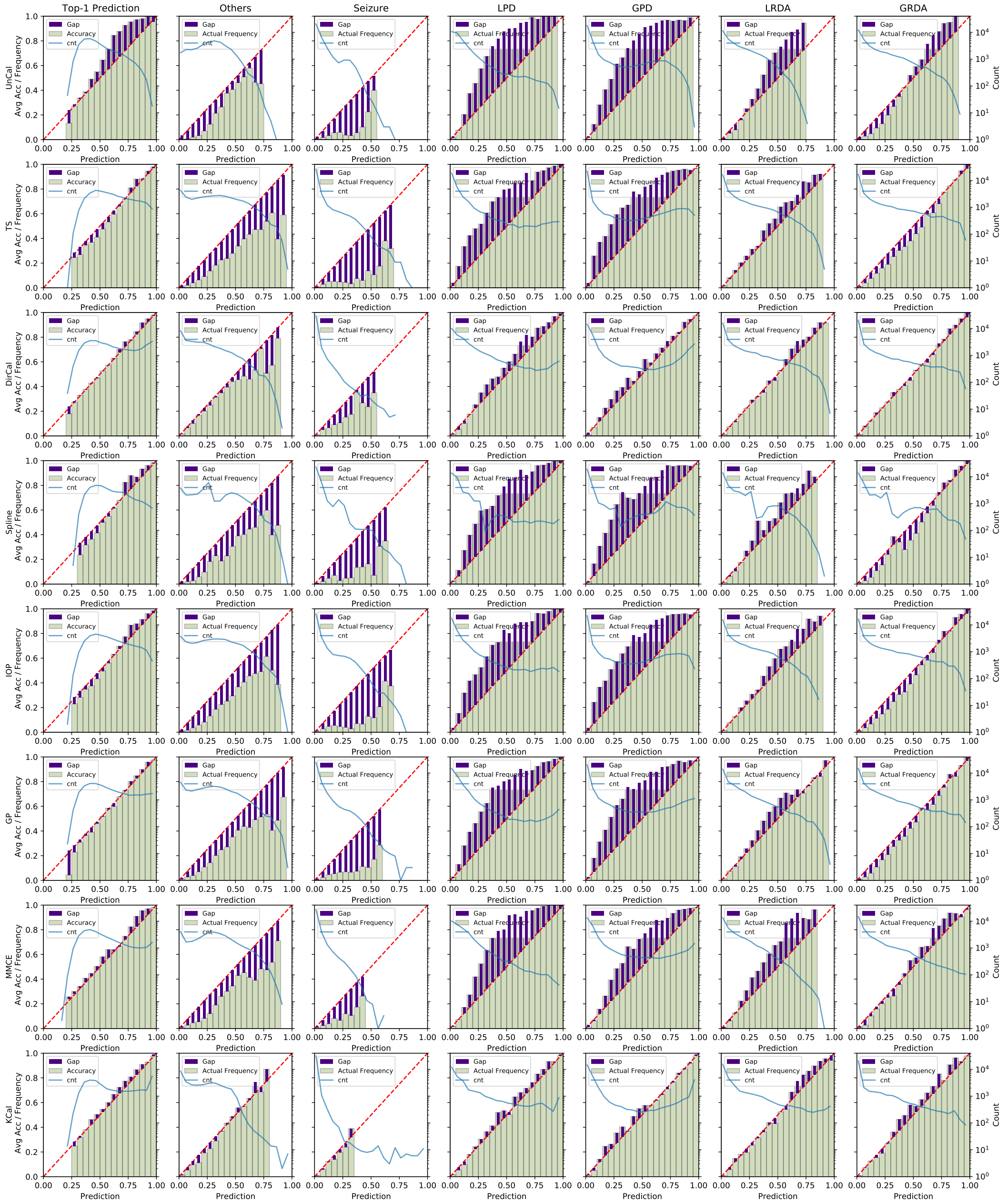

Figure 3: Reliability diagrams for the IIIC dataset.

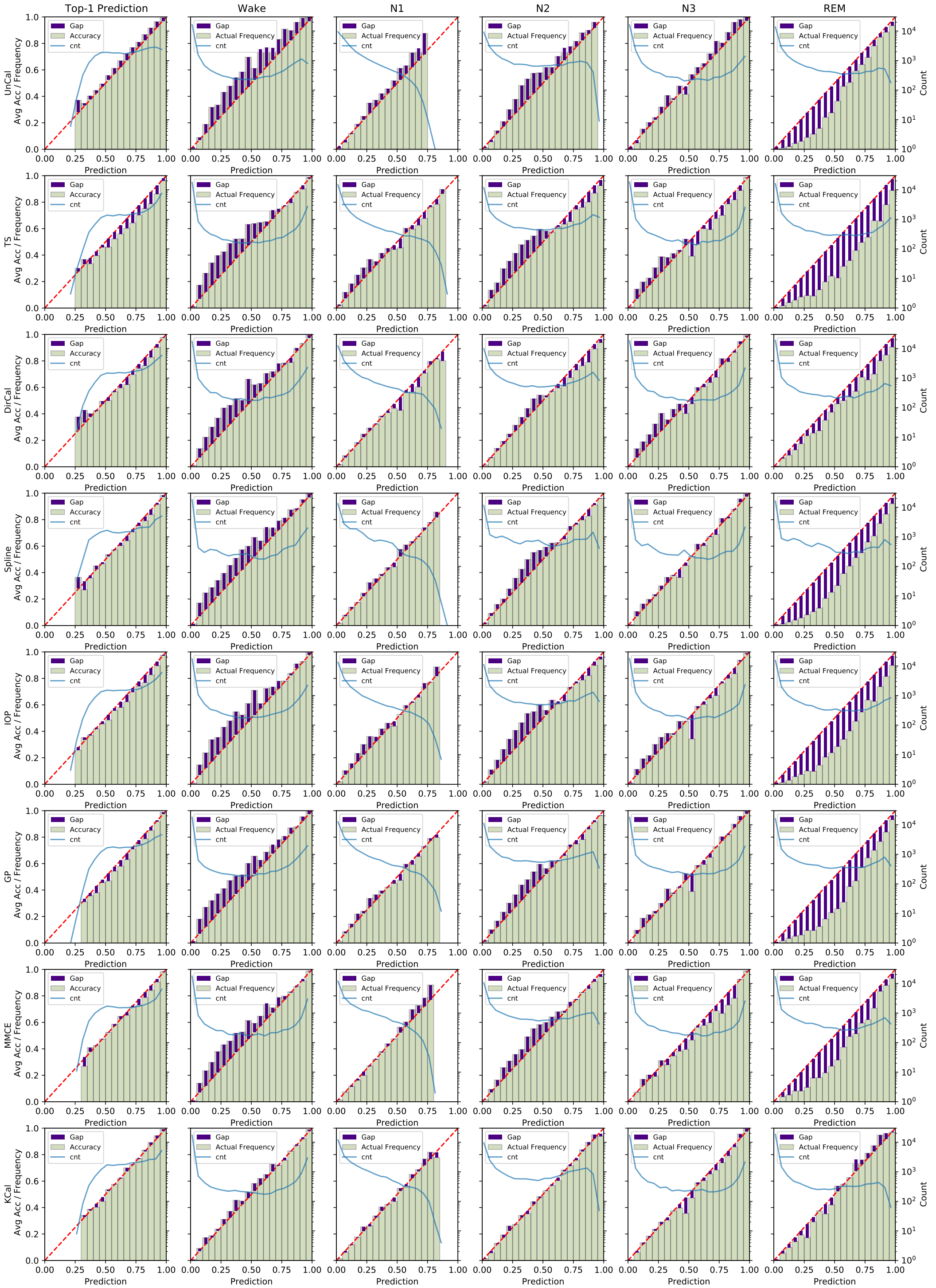

Figure 4: Reliability diagrams for the ISRUC dataset.

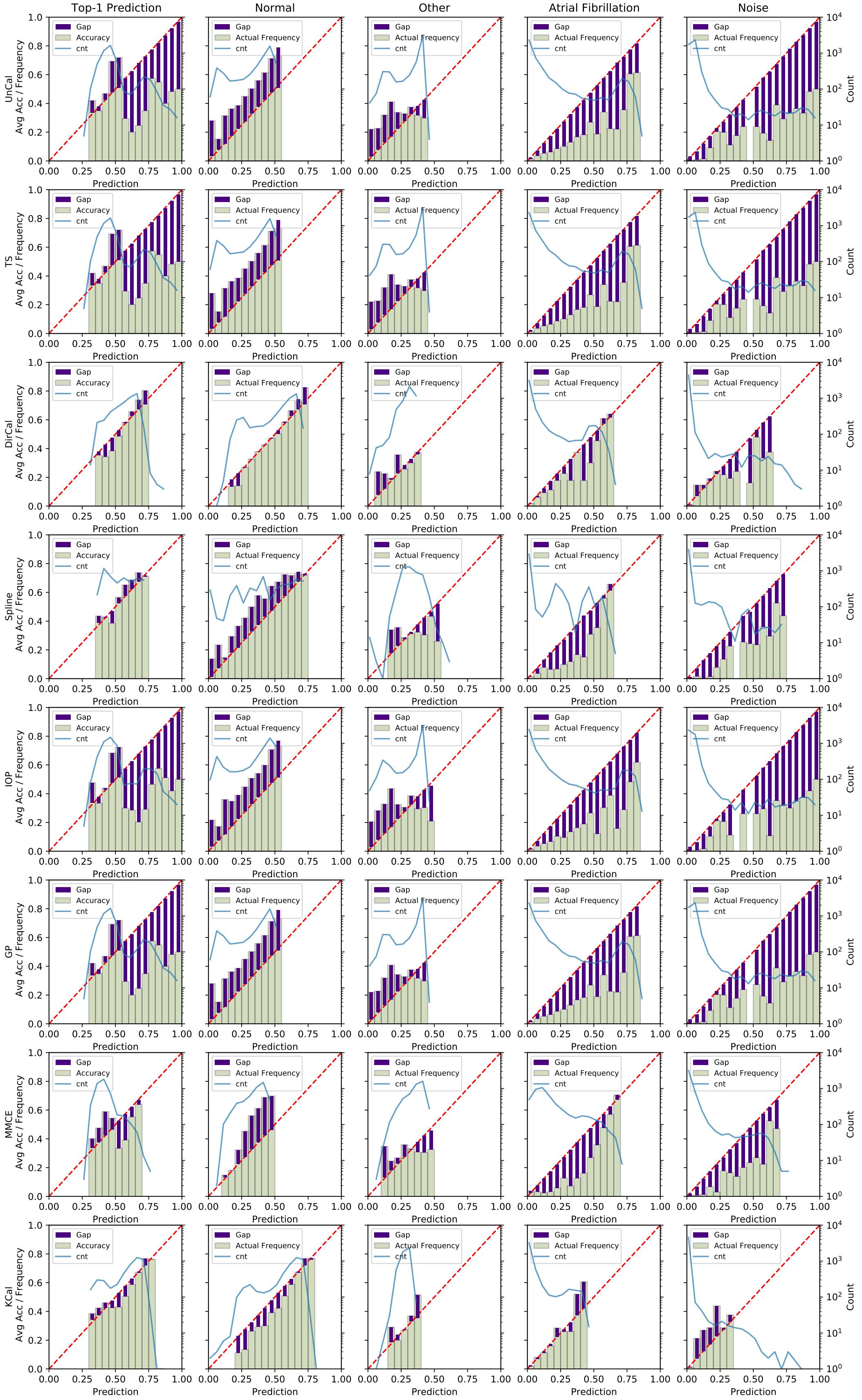

Figure 5: Reliability diagrams for the PN2017 dataset.

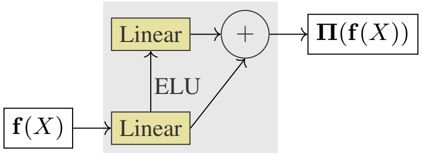

Figure 6: Structure of the learnable projection $\mathbf{\Pi}$ (in gray).

D Ablation Study: Linear Projection

A natural first architecture to try for $\mathbf{\Pi}$ is a simple linear layer. It is however not clear whether a linear projection can learn the best metric space due to its simplicity. We introduced a mild complexity by having two layers in $\mathbf{\Pi}$, yet the skip connection should help it learn well when a linear projection is the most desirable as well. We empirically compared both versions: KCal, with the architecture showed in Figure 6, and KCal-Linear, which only uses one linear layer with the same output dimension (d). Both $\mathbf{\Pi}$ normalized $\mathbf{f}(\cdot)$ automatically with a Batch Normalization layer. The results are in Table 12. As we can see, KCal is generally better than the linear version, but the gap is generally small. The additional computation time is smaller than 1x the computation time for KCal-Linear, because the second layer has only d^2 parameters rather than hd in the first layer ($h > d$). Both have negligible computation overhead compared with calling $\mathbf{f}$ (see Appendix F).

Table 12: Comparison between the architecture described in Figure 6 (KCal) and a simple linear projection with the same input and output dimensions (KCal-Linear). On average, KCal adapts to different datasets and architectures better than (KCal-Linear), although the performance is generally similar.

	Accuracy↑		CECE↓		ECE↓		Brier ↓	
	KCal	KCal-Lienar	KCal	KCal-Lienar	KCal	KCal-Lienar	KCal	KCal-Lienar
IIIC(pat)	**61.67±2.22**	61.51±2.46	**4.68±1.27**	4.68±1.41	**4.34±1.35**	4.48±1.99	19.33±0.78	**19.28±0.82**
IIIC	**66.32±0.21**	65.59±0.20	**2.03±0.26**	2.08±0.23	**2.62±0.59**	3.12±0.68	**17.54±0.10**	17.88±0.09
ISRUC(pat)	**76.13±0.89**	76.02±1.08	**3.82±1.24**	3.96±1.34	**2.78±1.25**	2.87±1.53	**14.97±0.29**	15.04±0.30
ISRUC	**77.45±0.16**	77.19±0.19	**1.90±0.28**	2.01±0.31	**1.36±0.41**	1.69±0.46	**14.28±0.08**	14.37±0.08
PN2017	**60.36±0.61**	60.15±0.56	4.25±1.26	**4.21±1.26**	**4.78±1.48**	5.41±1.14	**22.56±0.28**	22.69±0.32
C10 (ViT)	**98.98±0.09**	98.96±0.07	0.74±0.07	**0.72±0.06**	0.40±0.05	**0.31±0.06**	**0.75±0.05**	0.75±0.05
C10 (Mixer)	**98.14±0.06**	98.12±0.09	**1.17±0.10**	1.18±0.07	**0.59±0.09**	0.61±0.13	**1.34±0.04**	1.34±0.05
C100 (ViT)	92.37±0.15	**92.47±0.14**	**4.32±0.10**	4.37±0.08	1.50±0.32	**1.43±0.33**	5.01±0.08	**4.93±0.08**
C100 (Mixer)	87.55±0.16	**88.00±0.24**	**4.62±0.10**	4.73±0.12	3.07±0.49	**2.78±0.45**	7.61±0.09	**7.39±0.07**
SVHN (ViT)	**96.42±0.05**	96.36±0.06	**1.23±0.10**	1.32±0.08	**0.64±0.12**	0.65±0.09	**2.49±0.03**	2.49±0.03
SVHN (Mixer)	96.10±0.04	**96.13±0.04**	**1.40±0.08**	1.49±0.08	0.73±0.10	**0.61±0.09**	**2.68±0.03**	2.69±0.03

E Ablation Study: Using the classification logits

We empirically compared using the penultimate-layer embeddings and the predicted logits in Table 13. As we can see, KCal is generally better than the alternative that uses the logits.

Table 13: Comparison between using the penultimate layer embedding vs the prediction logits as the input to $\mathbf{\Pi}$ (KCal-Logits). Overall, KCal is significantly better than KCal-Logits, but KCal-Logits also has competitive performance.

	Accuracy↑		CECE↓		ECE↓		Brier ↓	
	KCal	KCal-Logits	KCal	KCal-Logits	KCal	KCal-Logits	KCal	KCal-Logits
IIIC(pat)	**61.67±2.22**	61.21±2.66	4.68±1.27	**4.26±1.30**	4.34±1.35	**4.02±1.51**	19.33±0.78	**19.07±0.77**
IIIC	**66.32±0.21**	65.26±0.20	**2.03±0.26**	2.11±0.27	**2.62±0.59**	2.77±0.37	**17.54±0.10**	17.90±0.05
ISRUC(pat)	**76.13±0.89**	75.57±1.02	**3.82±1.24**	3.95±1.44	2.78±1.25	**2.75±1.27**	**14.97±0.29**	15.30±0.31
ISRUC	**77.45±0.16**	76.75±0.12	**1.90±0.28**	1.97±0.32	**1.36±0.41**	1.62±0.48	**14.28±0.08**	14.60±0.09
PN2017	**60.36±0.61**	59.99±0.56	4.25±1.26	**4.13±1.22**	**4.78±1.48**	5.18±0.96	**22.56±0.28**	22.64±0.34
C10 (ViT)	**98.98±0.09**	98.94±0.06	**0.74±0.07**	0.79±0.07	**0.40±0.05**	0.43±0.05	**0.75±0.05**	0.79±0.04
C10 (Mixer)	**98.14±0.06**	98.11±0.06	**1.17±0.10**	1.21±0.07	0.59±0.09	**0.54±0.06**	**1.34±0.04**	1.37±0.04
C100 (ViT)	**92.37±0.15**	91.11±0.14	**4.32±0.10**	4.67±0.10	**1.50±0.32**	1.95±0.37	**5.01±0.08**	5.55±0.08
C100 (Mixer)	**87.55±0.16**	85.07±0.26	**4.62±0.10**	4.98±0.13	**3.07±0.49**	3.73±0.54	**7.61±0.09**	8.84±0.06
SVHN (ViT)	**96.42±0.05**	96.05±0.05	**1.23±0.10**	1.53±0.12	**0.64±0.12**	0.91±0.08	**2.49±0.03**	2.76±0.03
SVHN (Mixer)	**96.10±0.04**	95.90±0.05	**1.40±0.08**	1.65±0.11	**0.73±0.10**	0.88±0.09	**2.68±0.03**	2.84±0.03

F Ablation Study: Effect of d

To investigate the effect of d, we tried d =8, 16, 32, 64, and 128 and repeat the experiments. The performance and the inference time (overhead) can be found in Figure 7. The inference time depends on the size of the calibration set, which is specified in Section 4.

Generally speaking, we can only tell for sure that increasing d increases the overhead, although the overhead is always small compared with calling $\mathbf{f}$. The effect on other metrics, including accuracy, ECE and CEC, is not monotonic, and the best d probably depends on many factors.

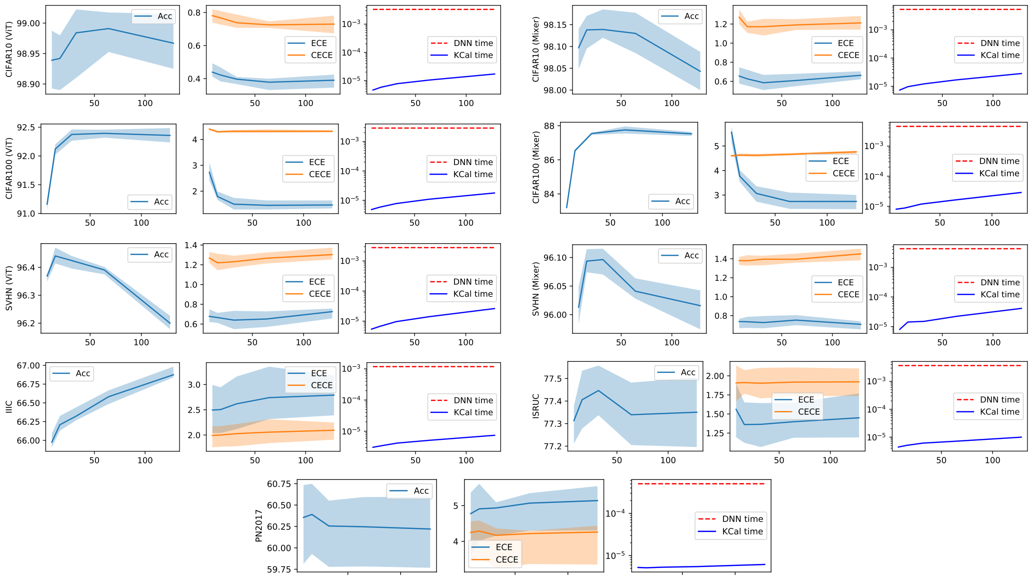

Figure 7: Change in performance and inference time if we we change d (the output embedding size of $\mathbf{\Pi}$. "DNN time" refers to the average time running $\mathbf{f}$ for one input x, and "KCal time" refers to the average time transforming $\mathbf{f}(x)$ to $\hat{\mathbf{p}}(x)$ using KCal. For Accuracy, ECE and CECE, the unit is percentage. The band represents the median 50% among 10 experiments. For time, the unit is second. Performance is not always improving as d increases, but a larger d naturally leads to larger overhead. It is however worth noting that in all experiment, the overhead ("KCal time") is negligible compared with the 'DNN time".

G COMPUTING BANDWIDTH

As suggested in the main text, although there is a bandwidth selection step that seemingly prevents KCal from efficiently updating predictions in an online manner, we could actually leverage Lemma 3.2 to compute b as opposed to actually performing cross-validation. To verify empirically that this is feasible in practice, we perform experiments where we vary the size of the calibration set, and plot the cross-validation-selected bandwidth b against the predicted value $\Theta(m^{-\frac{1}{d+4}})$. The results are in Figure 8.

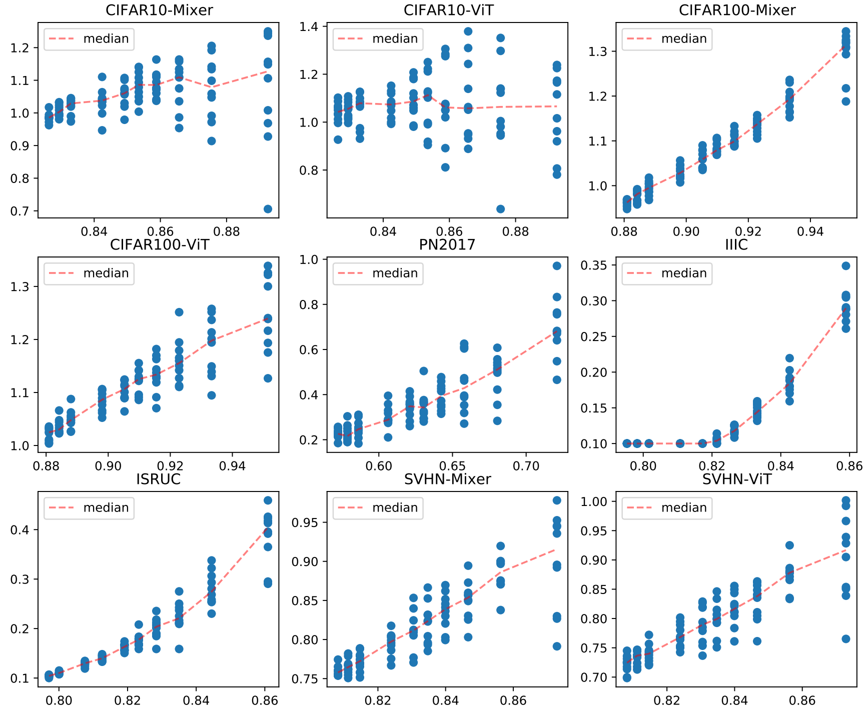

Figure 8: Empirically-selected bandwidth (b^*) on the y-axis, and predicted bandwidth ($\Theta(m^{-\frac{1}{d+4}})$) on the x-axis. For each calibration set size we have 10 experiments like in the main text, and we plot the scatter plot and median of the experiments. As expected, we see a nearly linear relationship for most data, except for IIIC, which exhibits a piece-wise linear pattern. This suggests that in practice, as new samples are added into the calibration set in an online manner, we could *compute* the bandwidth b and only re-do the cross validation sparingly.

If everything is perfect, we should see a linear relation in all plots, and we can use this relationship to compute b^* when we gradually add samples to the calibration set. It is clear that if we use the estimated constant in the $\Theta(\cdot)$ and the calibration set size (per class) m to set the bandwidth, we are still very close to the empirically selected value most of the time. In practice, this means that we only need to perform the actual cross validation occasionally, and predict the b^* in between. Note that from left to right, m decreases, so the optimal b^* increases and the variance increases greatly due to m being small. In practice, one might keep updating b^* using cross validation when m is small (and cross-validation takes very little time) and only compute b^* when m is already large.

While computation will give good estimates for b^* for most datasets, especially when m is large and the estimate of b^* is relatively stable (towards the left ends of plots), IIIC (and ISRUC to some extent) seems to show two different slopes. As m increases, from right to left, b^* seems to first decrease, and then stop decreasing. While a detailed analysis for this are beyond the scope of this paper, there are a few possible reasons.

1. First, and most importantly, the optimal bandwidth derived in Lemma 3.2 is "best" for estimating the density, f_k (in Eq. (4), not $\mathbb{P}\{\cdot|X\}$. b^* is however chosen according to the log-loss of the KDE classifier. As a result, the formula should be more relevant when K is large and the difference between $\hat{\mathbf{p}}_k(X)$ and $\mathbb{P}\{Y = k|X\}$ is essentially linear in $\hat{f}_k - f_k$ (as the denominator is much more accurate than the numerator). The experiment does support this point, since CIFAR100, with 100 classes, exhibits the clearest linear relationship.
2. Lemma 3.2 is not applicable if $f_{\mathbf{\Pi}\circ\mathbf{f}}$ violates the assumptions. For example, if $\mathbf{f}$ creates a discontinuity in the density, with a lot of data from different classes mapped to the same embedding. This means decreasing b might not decrease the bias term in Section A.2, and only increases variance. This could be what is happening in CIFAR10-ViT (with 99% accuracy) and in the left end of IIIC: decreasing b might not improve log-loss as we have exhausted the discriminative power of $\mathbf{f}$.

H Bandwidth Selection

In Section 3.4, we stated that we use Golden-Section search because we assume the cross entropy loss is convex in bandwidth b. While the convexity is expected from the bias-variance trade-off, we show in Figure 9 that this is indeed the case.

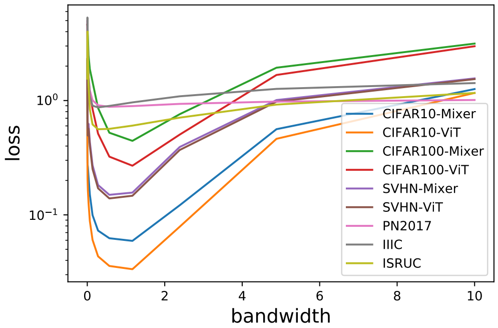

Figure 9: Change in cross validation loss used for the Golden-Section Search mentioned in Section 3.4 as a function of bandwidth. As we can see, the loss is indeed roughly convex in the bandwidth for all datasets.

I Effect of the size of the calibration set

In Figure 10, we plot the accuracy, Brier score, CECE and ECE as a function of $|\mathcal{S}_{\text{cal}}|$ for different datasets. As expected, as $|\mathcal{S}_{\text{cal}}|$ increases, the performance of KCal increases and then stabilizes.

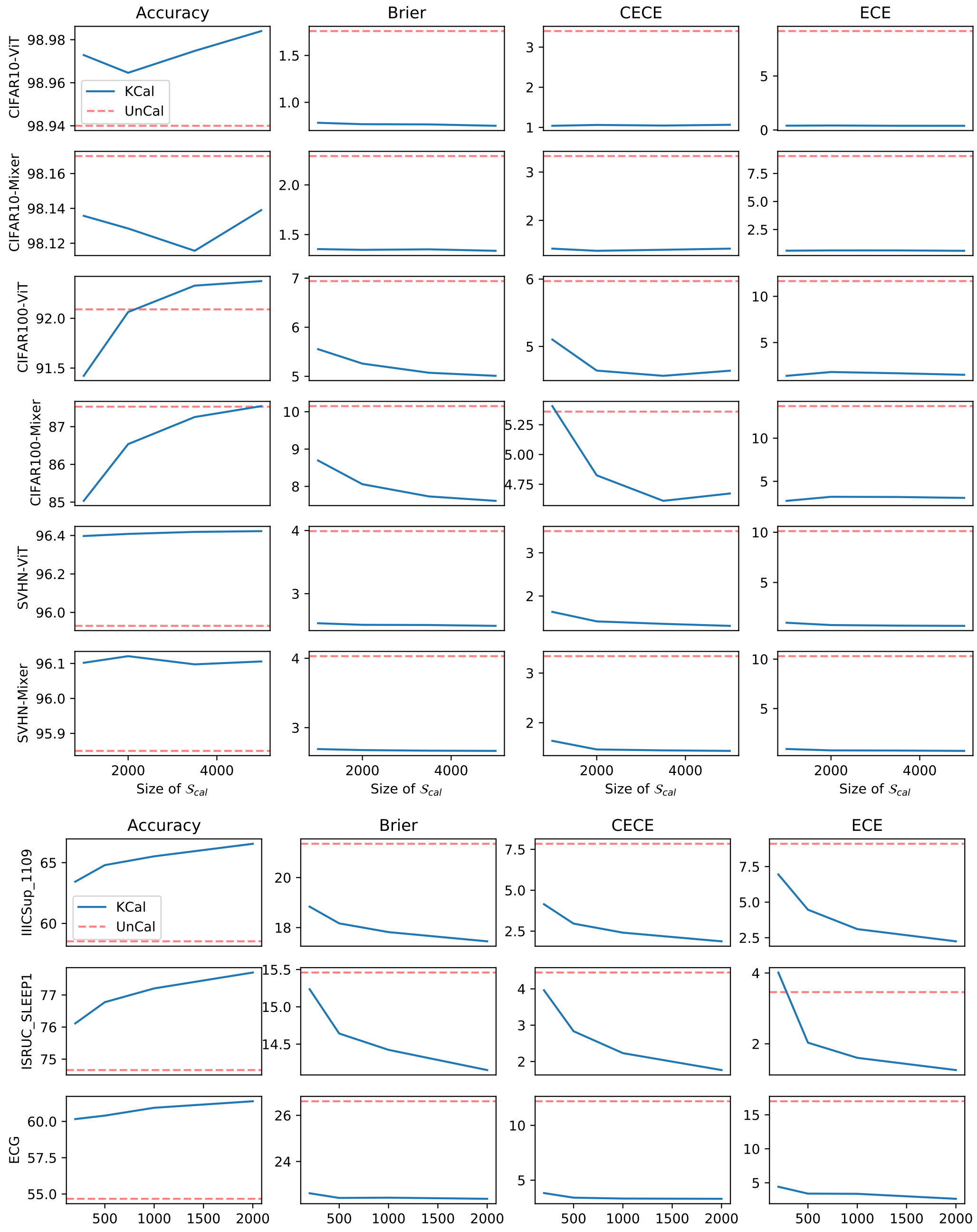

Figure 10: We change the size of $\mathcal{S}_{\text{cal}}$ and repeat the experiment for KCal. The red dashed line denotes `UnCal`. We see that, as expected, all metrics improves as the size of the calibration set increases, and the performance is generally stable.