A Expressing Popular Forms of Calibration as Distribution Matching

A.1 Calibration in Regression

Here, we show how the forms of calibration listed in Table 1 can be expressed in terms of conditional distribution matching. Note that many of these forms of calibration are sensible in the classification setting as well (e.g., marginal, group, and decision calibration).

Quantile Calibration Let Y be a continuous random variable. A forecaster is quantile calibrated if

$$\mathbb{P}\left(Q_{Y|X}(Y) \leq t\right) = t, \quad \forall t \in [0,1]. \tag{6}$$

This is equivalent to stating that $Q_{Y|X}(Y)$ is uniform on the interval $[0,1]$. For continuous random variables, the probability integral transform $P_{Y|X}(Y)$ is uniform on the interval $[0,1]$. Thus, quantile calibration can be written as $Q_{Y|X}(Y) \triangleq P_{Y|X}(Y)$, where $\triangleq$ denotes equality in distribution.

Threshold Calibration Let Y be a continuous random variable. A forecaster satisfies threshold calibration [39] if $\mathbb{P}(Q_{Y|X}(Y) \leq t \mid Q_{Y|X}(y) \leq c) = t$ for all $t \in [0,1]$, $c \in [0,1]$, and $y \in \mathcal{Y}$. This states that $Q_{Y|X}(Y)$ is uniform, conditioned on the event $Q_{Y|X}(y) \leq c$. Noting that $P_{Y|X}(Y)$ is uniform, the constraint is then equivalent to the following:

$$Q_{Y|X}(Y) \triangleq P_{Y|X}(Y) \mid \mathbb{1}\left\{Q_{Y|X}(y) \leq c\right\}, \quad \forall c \in [0,1], y \in \mathcal{Y} \tag{7}$$

Marginal Calibration A forecaster is marginally calibrated [11] if $\mathbb{E}[Q_{Y|X}(y)] = P_Y(y)$ for all $y \in \mathcal{Y}$. Recall that the distribution of $\widehat{Y}$ is defined by $(\widehat{Y} \mid X = x) \sim Q_{Y|x}$. Thus, the marginal distribution of $\widehat{Y}$ is given by $\widehat{Y} \sim \mathbb{E}[Q_{Y|X}]$. So marginal calibration states that $\widehat{Y}$ and Y have the same marginal cdf or, equivalently, $\widehat{Y} \triangleq Y$.

Decision Calibration Zhao et al. [52] introduce the concept of $\mathcal{L}^K$ decision calibration for multi-class classification, where $\mathcal{Y} = \{0,1\}^C$ is the set of onehot vectors over C classes.

Definition A.1 ($\mathcal{L}^K$-Decision Calibration, Zhao et al. [52]). Let $\mathcal{L}^K$ be the set of all loss functions with K actions $\mathcal{L}^K = \{\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}, |\mathcal{A}| = K\}$, we say that a prediction $Q_{Y|X}$ is $\mathcal{L}^K$-decision calibrated (with respect to $P_{Y|X}$) if $\forall \ell \in \mathcal{L}^K$ and $\delta \in \Delta_{\mathcal{L}^K}$

$$\mathbb{E}_X \mathbb{E}_{\widehat{Y} \sim Q_{Y|X}}[\ell(\widehat{Y}, \delta(Q_{Y|X}))] = \mathbb{E}_X \mathbb{E}_{Y \sim P_{Y|X}}[\ell(Y, \delta(Q_{Y|X}))] \tag{8}$$

where $\Delta_{\mathcal{L}^K}$ is the set of all Bayes decision rules over loss functions with K actions.

They show that $\mathcal{L}^K$ decision calibration can equivalently be expressed as requiring that $\mathbb{E}[(\widehat{Y} - Y)\mathbb{1}\{\delta(Q_{Y|X}) = a\}] = 0$ for all $\delta \in \Delta_{\mathcal{L}^K}$. We can rewrite this as

$$\mathbb{E}[\widehat{Y} | \delta(Q_{Y|X}) = a] = \mathbb{E}[Y | \delta(Q_{Y|X}) = a], \quad \forall \delta \in \Delta_{\mathcal{L}^K}, a \in \mathcal{A} \tag{9}$$

note that $\mathbb{E}[Y | \delta(Q_{Y|X}) = a]$ gives the conditional distribution of $(Y \mid \delta(Q_{Y|X}) = a)$ due to the onehot representation of Y. Thus, it is equivalent to require that

$$\left(\widehat{Y} \triangleq Y\right) \mid \delta(Q_{Y|X}), \quad \forall \delta \in \Delta_{\mathcal{L}^K} \tag{10}$$

where $\triangleq$ denotes equality in distribution.

Group Calibration Group calibration [20] says that forecasts are marginally calibrated for each subgroup (e.g., subgroups defined by gender or age). We can write this as $\widehat{Y} \triangleq Y \mid \text{Group}(X)$, where Group($X$) is a variable indicating which subgroup includes the feature vector X.

Distribution Calibration A forecaster satisfies distribution calibration [42] if for all possible forecasts Q_0 and labels $y \in \mathcal{Y}$, we have $\mathbb{P}\left(Y \leq y \mid Q_{Y|X} = Q_0\right) = Q_0(y)$. This is equivalent to requiring $P_{Y|X}(y) \mid Q_{Y|X} = Q_{Y|X}(y)$ for all $y \in \mathcal{Y}$. Since the cdfs take the same value at all $y \in \mathcal{Y}$, the random variables Y and $\widehat{Y}$ have the same distribution, giving us that $\widehat{Y} \triangleq Y \mid Q_{Y|X}$.

Individual Calibration Let Y be a real-valued continuous random variable. Individual calibration [51] states that $\mathbb{P}\left(Q_{Y|x}(Y) \le t\right) = t$, for all $t \in [0,1]$ and $x \in \mathcal{X}$. Letting $U \sim \mathrm{Unif}(0,1)$, we can write this as

$$Q_{Y|X}(Y) \triangleq P_{Y|X}(Y) \mid X \tag{11}$$

$$Q^{-1}_{Y|X}(Q_{Y|X}(Y)) \triangleq Q^{-1}_{Y|X}(P_{Y|X}(Y)) \mid X \tag{12}$$

$$Y \triangleq Q^{-1}_{Y|X}(U) \mid X \tag{13}$$

$$Y \triangleq \widehat{Y} \mid X \tag{14}$$

Local Calibration A forecaster is locally calibrated [27] if $Q_{Y|X}$ and $P_{Y|X}$ are equal, on average, in any neighborhood of the feature space. Similarity in feature space is defined by a kernel $k(x, x')$. In regression, local calibration for quantiles says that

$$\frac{\mathbb{E}[\mathbb{1}\left\{Q_{Y|X}(Y) \le t\right\} k(X,x)]}{\mathbb{E}[k(X,x)]} = t, \quad \forall t \in [0,1], x \in \mathcal{X}. \tag{15}$$

This can equivalently be stated as

$$\mathbb{E}[\mathbb{1}\left\{Q_{Y|X}(Y) \le t\right\} k(X,x)] = \mathbb{E}[\mathbb{1}\left\{P_{Y|X}(Y) \le t\right\} k(X,x)], \quad \forall t \in [0,1], x \in \mathcal{X} \tag{16}$$

or, in terms of the canonical feature mapping,

$$\mathbb{E}[\mathbb{1}\left\{Q_{Y|X}(Y) \le t\right\} \langle \phi(X), \phi(x)\rangle] = \mathbb{E}[\mathbb{1}\left\{P_{Y|X}(Y) \le t\right\} \langle \phi(X), \phi(x)\rangle], \quad \forall t \in [0,1], x \in \mathcal{X}. \tag{17}$$

Thus, it is sufficient for $\mathbb{E}[Q_{Y|X}]$ and $\mathbb{E}[P_{Y|X}]$ to have the same distribution conditioned on the feature mapping $\phi(X)$. This can be written succinctly as

$$Y \triangleq \widehat{Y} \mid \phi(X) \tag{18}$$

A.2 Calibration in Classification

Here, we express three popular forms of calibration from the classification literature—namely, canonical, top-label, and marginal calibration [44, Eq (1-3)]—in terms of distribution matching constraints. For the entirety of this section, Y is a discrete random variable taking values in the set $\mathcal{Y} = \{1, 2, \ldots, m\}$, and $q_{Y|x}$ is the probability mass function of the forecast given the features $X = x$.

Canonical Calibration A forecaster satisfies canonical calibration if $\mathbb{P}\left(Y = y \mid q_{Y|X}\right) = q_{Y|X}(y)$ for all $y \in \mathcal{Y}$. This says that the label follows the same distribution under the forecast and the true distribution, conditioned on $q_{Y|X}$. In other words, $Y \triangleq \widehat{Y} \mid q_{Y|X}$.

Top-Label Calibration A forecaster satisfies top-label calibration if $\mathbb{P}\left(Y = Y^* \mid q_{Y|X}(Y^*)\right) = q_{Y|X}(Y^*)$, where $Y^* := \arg\max_{y \in \mathcal{Y}} q_{Y|X}(y)$ is the mode of the forecast. This equation can be rewritten in terms of the indicators

$$\mathbb{E}\left[\mathbb{1}\left\{Y = Y^*\right\} \mid q_{Y|X}(Y^*)\right] = \mathbb{E}\left[\mathbb{1}\{\widehat{Y} = Y^*\} \mid q_{Y|X}(Y^*)\right]$$

and the indicator variables are equal in expectation if and only if they are equal in distribution. Thus, top-label calibration is equivalent to $\mathbb{1}\left\{Y = Y^*\right\} \triangleq \mathbb{1}\{\widehat{Y} = Y^*\} \mid q_{Y|X}(Y^*)$.

Marginal Calibration A forecaster satisfies marginal calibration if $\mathbb{P}\left(Y = y \mid q_{Y|X}(y)\right) = q_{Y|X}(y)$ for all $y \in \mathcal{Y}$. Similar to top-label calibration, this can be written as

$$\mathbb{E}\left[\mathbb{1}\left\{Y = y\right\} \mid q_{Y|X}(y)\right] = \mathbb{E}\left[\mathbb{1}\{\widehat{Y} = y\} \mid q_{Y|X}(y)\right]$$

and the indicator variables are equal in expectation if and only if they are equal in distribution. Thus, marginal calibration can be written as $\mathbb{1}\left\{Y = y\right\} \triangleq \mathbb{1}\{\widehat{Y} = y\} \mid q_{Y|X}(y)$, for all $y \in \mathcal{Y}$.

B Experimental Setup and Additional Results

B.1 Reproducibility

In order to reproduce the experiments and results obtained in this paper, we provide details about the hyperparameter tuning and required computational resources.

Hyperparameter Search: We use TPES [3] to perform hyperparameter optimization. For all experiments, we vary:

- Layer sizes between 32 and 512
- RBF kernel bandwidths between 0.001 and 200
- Batch sizes between 16 and 512, with and without batch normalization
- Learning rates between 10^{-7} and 10^{-1}.
- The loss mixture weight λ (as in $\mathrm{NLL} + \lambda \cdot \mathrm{MMD}$ and $\mathrm{XE} + \lambda \cdot \mathrm{MMD}$) between 0.1 and 1000.

For each dataset, we test 100 hyperparameter settings for each training objective for regression $\{\mathrm{NLL}, \mathrm{NLL} + \mathrm{MMD}\}$, and classification setting $\{\mathrm{XE}, \mathrm{XE} + \mathrm{MMCE}, \mathrm{XE} + \mathrm{KDE}, \mathrm{XE} + \mathrm{MMD}\}$. For each run, we enable early-stopping if the validation loss does not improve for more than 50 epochs. In the regression setting, we pick the best performing run based on the validation NLL. For classification, we select the best performing run based on the validation accuracy, with validation ECE used for break ties. For each model and dataset, the best performing model is then re-run with 50 random seeds to gather information about standard errors and statistical significance. To capture variability across different seeds, we report the standard error of the mean in Tables 3, 4, and 5.

Kernel Bandwidth We select the RBF kernel bandwidth for training on each dataset using the aforementioned hyperparameter optimization. For each dataset in Tables 4 and 5, the KDE metric we report is for the bandwidth selected through a held-out validation set, scaled so that the error is of a comparable order of magnitude across datasets (since multiplying a kernel by a positive scalar preserves the kernel properties).

Baselines In the classification setting, we compare our approach to two state-of-the-art trainable calibration methods, MMCE [26] and KDE [37]. For MMCE, we follow the same experimental setup as the official author implementation, where the loss mixture weight λ and the Laplacian kernel bandwidth are chosen via a held-out validation set. For KDE, we use the L_1 canonical calibration using the ECE^{KDE} estimator, calculated according to Equation (9) from Popordanoska et al. [37]. We follow the same experimental setup as the official author implementation, where the kernel bandwidth is chosen using leave-one-out-likelihood (LOO MLE), and the loss mixture weight λ is chosen via a held-out validation set.

Computational Requirements: All experiments were conducted on a single CPU machine (Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz), utilizing 8 cores per experiment. When run on a CPU, the training time for a single model ranges from 5 to 15 minutes, depending on the dataset size. It is possible to perform a full hyperparameter sweep using this setup in $\sim$24 hours.

To accelerate training, we have also run some experiments using an 11GB NVIDIA GeForce GTX 1080 Ti. When run on a GPU, the training time for a single model ranges from 2 to 5 minutes, depending on the dataset size, and a hyperparameter sweep takes $\sim$10 hours.

US Agriculture Dataset: We introduce a new CROP-YIELD dataset for predicting yearly wheat crop yield from geospatial weather data for different counties across the US. We use NOAA database which contains weather data from thousands of stations across the United States. For each county, we track the weather sequence of each year into a few summary statistics for each month (average/maximum/minimum temperatures, precipitation, cooling/heating degree days). To study how climate change and location affect crop yield, we match crop yield data from the NASS dataset to weather stations in the NOAA database, and learn a function to predict crop yield from weather data. We specifically considered the crop yield of wheat across different counties in the US between 1990-2007. Since crop yield data for 1995 is the least sparse, we hold out data from 1995 for visualization, and train our forecasters on the remaining data.

B.2 Additional Experimental Results

Table 5: Performance on classification tasks, comparing MMCE regularization [26], KDE regularization [37], and MMD regularization (*Ours*), alongside a standard cross-entropy (XE) loss. Temperature scaling is used for post-hoc recalibration of all methods. n is the number of examples in the dataset, d is the number of features, and m is the number of classes.

Dataset	Training Objective **with Temp. Scaling**	Accuracy ↑	ECE ↓	Entropy ↓	KCE ↓
BREAST-CANCER	XE	95.372 ± 0.160	0.105 ± 0.031	0.056 ± 0.007	0.028 ± 0.005
$n = 569$	XE + MMCE	94.770 ± 0.147	**0.052 ± 0.001**	0.011 ± 0.001	**0.019 ± 0.000**
$d = 30$	XE + ECE KDE	94.351 ± 0.163	0.074 ± 0.018	0.010 ± 0.001	0.022 ± 0.003
$m = 2$	XE + MMD (*Ours*)	**95.789 ± 0.060**	**0.052 ± 0.000**	**0.004 ± 0.000**	**0.019 ± 0.000**
HEART-DISEASE	XE	55.904 ± 0.196	0.325 ± 0.003	1.116 ± 0.008	0.077 ± 0.000
$n = 921$	XE + MMCE	60.787 ± 0.208	0.262 ± 0.002	0.927 ± 0.004	0.067 ± 0.000
$d = 23$	XE + ECE KDE	50.036 ± 2.801	0.276 ± 0.011	1.192 ± 0.033	0.082 ± 0.002
$m = 5$	XE + MMD (*Ours*)	**61.516 ± 0.255**	**0.252 ± 0.003**	**0.860 ± 0.003**	**0.066 ± 0.000**
ONLINE-SHOPPERS	XE	89.816 ± 0.037	0.130 ± 0.000	0.227 ± 0.001	**0.050 ± 0.000**
$n = 12330$	XE + MMCE	89.933 ± 0.036	0.128 ± 0.000	0.220 ± 0.000	0.051 ± 0.000
$d = 28$	XE + ECE KDE	**90.019 ± 0.034**	0.125 ± 0.000	0.216 ± 0.001	**0.050 ± 0.000**
$m = 2$	XE + MMD (*Ours*)	89.976 ± 0.031	**0.126 ± 0.000**	**0.214 ± 0.001**	**0.050 ± 0.000**
DRY-BEAN	XE	92.071 ± 0.025	0.102 ± 0.000	0.206 ± 0.001	**0.011 ± 0.000**
$n = 13612$	XE + MMCE	92.772 ± 0.035	0.092 ± 0.000	0.188 ± 0.001	**0.011 ± 0.000**
$d = 16$	XE + ECE KDE	92.760 ± 0.037	0.091 ± 0.000	0.190 ± 0.000	**0.011 ± 0.000**
$m = 7$	XE + MMD (*Ours*)	**92.894 ± 0.035**	**0.086 ± 0.000**	**0.187 ± 0.001**	**0.011 ± 0.000**
ADULT	XE	84.528 ± 0.040	0.188 ± 0.000	**0.330 ± 0.000**	0.022 ± 0.000
$n = 32561$	XE + MMCE	84.203 ± 0.042	0.190 ± 0.000	0.335 ± 0.000	0.022 ± 0.000
$d = 104$	XE + ECE KDE	84.187 ± 0.045	0.177 ± 0.001	0.338 ± 0.001	0.023 ± 0.000
$m = 2$	XE + MMD (*Ours*)	**84.565 ± 0.035**	**0.174 ± 0.000**	0.334 ± 0.000	**0.020 ± 0.000**

Table 6: Performance versus the number of simulated samples per forecast used for the plug-in MMD estimate in regression tasks. All other hyperparameters are held constant, including the number of training steps. Increasing the number of samples gives a better estimate of the MMD objective, generally leading to better performance at the cost of additional compute.

	CRIME			BLOG			MEDICAL-EXPENDITURE		
# Samples	NLL	QCE	DCE	NLL	QCE	DCE	NLL	QCE	DCE
1	-0.703	0.198	0.088	0.952	0.379	4.171	1.546	0.071	0.448
2	-0.755	0.188	0.133	0.959	0.444	4.228	1.535	0.07	0.427
5	-0.72	0.154	0.043	0.96	0.395	4.09	1.539	0.071	**0.419**
10	-0.777	0.154	0.06	0.85	0.416	4.077	1.54	0.072	0.459
20	-0.772	0.157	0.054	0.835	0.415	4.031	1.538	**0.063**	0.431
50	-0.779	**0.150**	**0.041**	0.847	0.432	4.016	**1.531**	0.065	0.447
100	-0.781	0.153	0.043	**0.829**	0.386	**3.978**	1.536	0.064	0.449
200	**-0.782**	0.151	0.043	**0.829**	**0.373**	**3.978**	1.535	0.064	0.442

	SUPERCONDUCTIVITY			FB-COMMENT		
# Samples	NLL	QCE	DCE	NLL	QCE	DCE
1	3.369	0.053	0.182	0.637	0.268	3.15
2	3.35	0.062	0.197	0.477	0.294	3.14
5	3.311	0.038	0.216	**0.409**	0.276	3.146
10	3.333	0.036	0.208	0.59	0.278	**3.102**
20	3.298	0.055	0.262	0.569	0.258	3.119
50	3.345	0.041	0.208	0.479	**0.223**	3.14
100	3.318	0.036	**0.181**	0.469	0.23	3.149
200	**3.291**	**0.034**	0.182	0.458	0.231	3.137

B.3 Experiment: Decision Calibration

As an illustrative example of how trainable calibration metrics can improve calibration for a specific decision-making problem, we consider a common real-world decision problem, requiring us to make an action whose utility depends only on whether the label y is greater or less than a particular threshold c. Concretely, the decision loss $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$, for an action $a \in \{-1, +1\}$ and a label $y \in \mathcal{Y}$, is defined as $\ell(a, y) = \mathbb{1}\{a \neq \text{sign}(y - c)\}$.

Setup In order to achieve decision calibration, we tailor our kernels k_X, k_Y to closely follow the decision problem. More concretely, we choose a kernel function that penalizes predicted labels which are on the wrong side of c, namely $k_Y(y, y') = \tanh(y - c) \cdot \tanh(y' - c)$. Notice that $k_Y(y, y') \approx +1$ when $\text{sign}(y - c) = \text{sign}(y - c)$, and -1 otherwise. Thus, the calibration metric gives the model training feedback that is well aligned with the decision problem. To encourage decision calibration across all features, we set k_X to be the universal RBF kernel.

As part of this experiment, we investigate whether tailoring an MMD kernel to the decision problem will improve decision calibration over using a universal RBF kernel for both k_X, k_Y.

Metric To evaluate the effectiveness of methods, we compute the standard Decision Calibration Error (DCE) across all regression datasets, defined as

$$\text{DCE}^2 = \sum_{a \in \mathcal{A}} \left|\left| \mathbb{E}_P[\ell(Y, a)] - \mathbb{E}_X \mathbb{E}_{\widehat{Y} \sim Q_{Y|X}}[\ell(\widehat{Y}, a)] \right|\right|_2^2, \tag{19}$$

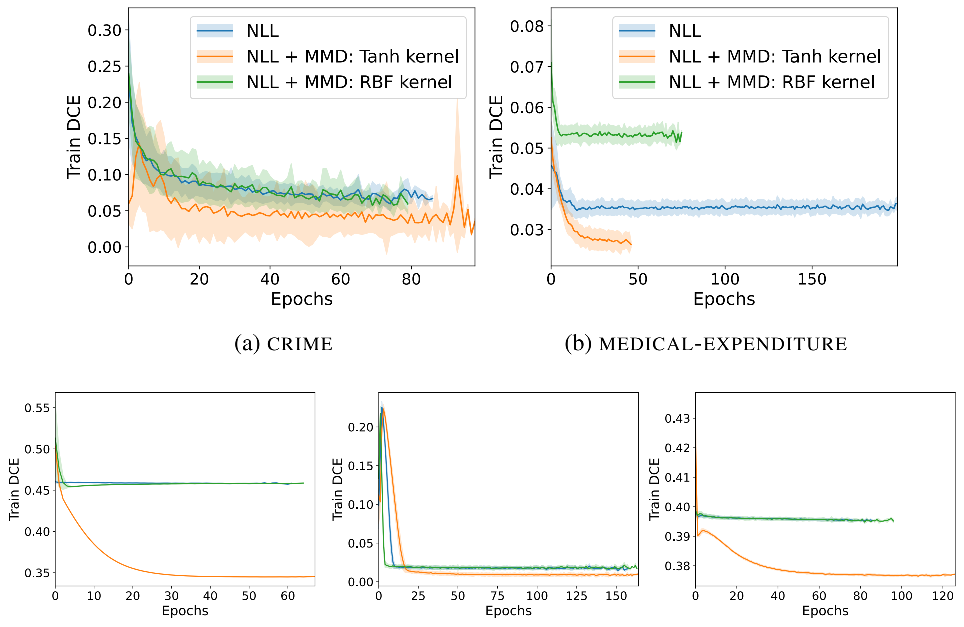
where P is the true distribution over $\mathcal{X} \times \mathcal{Y}$, and $Q_{Y|X}$ is a forecaster.

Additional Results: Choice of kernel We present two sets of results. The first set of results is shown in Table 7, comparing models trained using our objective (NLL + MMD), where one uses a kernel k_Y that is tailored to the decision problem ($\tanh$), while the other uses a universal RBF kernel. As described in Section 6.3, we observe that using the $\tanh$ kernel clearly achieves better decision calibration across all dataset, with minimal degradation in sharpness, i.e. NLL. This demonstrates that we can improve decision calibration using our trainable calibration metrics by tailoring our kernels to closely follow the decision problem.

The second set of results, shown in Figure 2, further demonstrates the relationship between our trainable calibration metrics and decision calibration throughout model training. By tailoring our kernels to closely follow the decision problem, we observe that decision calibration is optimized throughout training. In contrast, using objectives that are agnostic to the decision problem (i.e. NLL, or MMD with RBF kernels) will lead to noticeably worse decision calibration, with little to no improvement throughout training. The same trend is observed on a held-out validation set, as shown in Figure 3.

Dataset	**Kernel** (k_Y)	NLL ↓	Decision Cal. ↓
CRIME ($n = 1992, d = 102$)	RBF	**-0.778 ± 0.008**	0.042 ± 0.011
	tanh	-0.657 ± 0.010	**0.027 ± 0.006**
BLOG ($n = 52397, d = 280$)	RBF	0.957 ± 0.008	4.051 ± 0.002
	tanh	**0.934 ± 0.007**	**3.052 ± 0.001**
MEDICAL-EXPENDITURE ($n = 33005, d = 107$)	RBF	**1.530 ± 0.000**	0.447 ± 0.002
	tanh	1.532 ± 0.000	**0.438 ± 0.002**
SUPERCONDUCTIVITY ($n = 21264, d = 81$)	RBF	**3.269 ± 0.012**	0.182 ± 0.003
	tanh	3.311 ± 0.009	**0.127 ± 0.003**
FB-COMMENT ($n = 40949, d = 53$)	RBF	0.605 ± 0.004	3.131 ± 0.003
	tanh	**0.598 ± 0.003**	**2.342 ± 0.003**

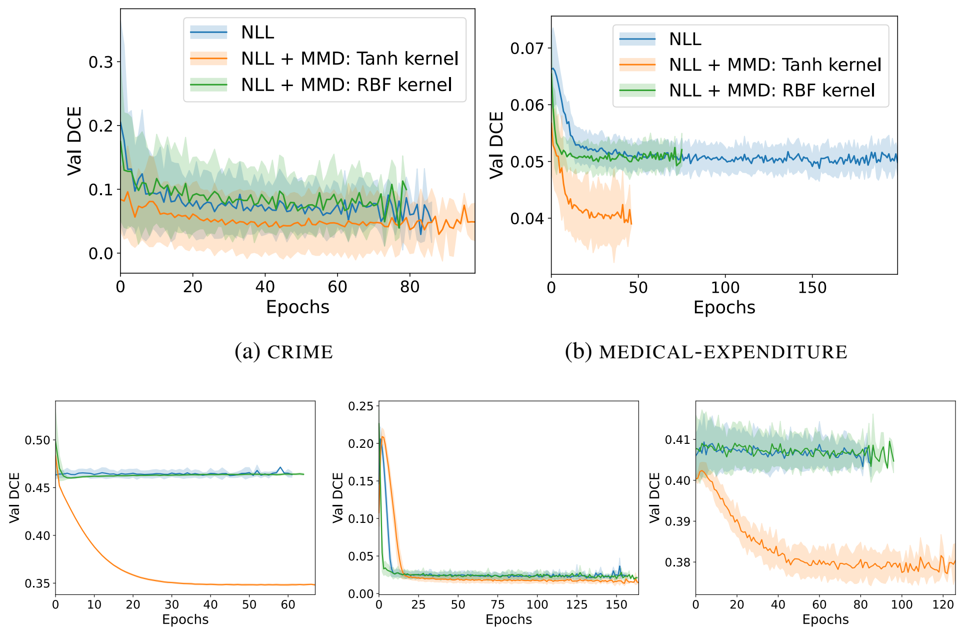
Table 7: Comparison of model performance trained with our training objective (NLL + MMD), using different kernels for k_Y in the MMD kernel. The RBF kernel is a commonly-used universal kernel, while the $\tanh$ kernel is specifically designed to match the decision task for decision calibration. All metrics are reported on the test set.



(a) CRIME (b) MEDICAL-EXPENDITURE

(c) BLOG (d) SUPERCONDUCTIVITY (e) FB-COMMENT

Figure 2: We visualize the Decision Calibration Error (DCE) evaluated throughout training on the training data, for a Gaussian forecaster trained using different objectives. Our method, by tailoring the kernels to a specific decision problem, achieves the best DCE among baselines across all datasets. Using our method, we observe continuous improvement in DCE throughout training. Error bars denote the standard deviation computed over 50 random trials.



(a) CRIME (b) MEDICAL-EXPENDITURE

(c) BLOG (d) SUPERCONDUCTIVITY (e) FB-COMMENT

Figure 3: We visualize the Decision Calibration Error (DCE) evaluated throughout training on held-out validation data. Our method, by tailoring the kernels to a specific decision problem, achieves the best DCE among baselines across all datasets. Error bars denote the standard deviation computed over 50 random trials.