Local Calibration: Metrics and Recalibration (Supplementary Material)

Rachel Luo*[1] Aadyot Bhatnagar*[2] Yu Bai[2] Shengjia Zhao[1] Huan Wang[2] Caiming Xiong[2] Silvio Savarese[1,2] Stefano Ermon[1,2] Edward Schmerling[1] Marco Pavone[1]

[1]Stanford University, Stanford, California, USA
[2]Salesforce AI Research, Palo Alto, California, USA

A MODEL ARCHITECTURE, TRAINING, AND OTHER HYPERPARAMETERS

For ImageNet and CelebA, we compute the ECE, MCE, and LCE using 15 equal-width confidence bins. For the UCI communities and crime dataset, we use 5 equal-width bins because the dataset is much smaller (500 datapoints for recalibration). These numbers of bins represent a good tradeoff between bias and variance in estimating the relevant calibration errors. We also ran some initial experiments with equal-mass binning, but found that the results were very similar to those obtained with equal-width binning.

A.1 IMAGENET

For all experiments with the ImageNet dataset, we used the pre-trained ResNet-50 model from the PyTorch `torchvision` package as our classifier. To calculate the LCE and apply LoRe, we used pre-trained Inception-v3 features, applying either t-SNE to reduce their dimension to 3 or PCA to reduce their dimension to 50, as a feature representation for the kernel.

A.2 UCI COMMUNITIES AND CRIME

For all experiments with the UCI communities and crime dataset, we used a 3-hidden-layer dense neural network as our base classifier. Each hidden layer had a width of 100 and was followed by a Leaky ReLU activation. We applied dropout with probability 0.4 after the final hidden layer. We trained the model using the Adam optimizer with a batch size of 64 and a learning rate of 3×10^{-4} until the validation accuracy stopped improving. All other hyperparameters were PyTorch defaults. Training was done locally on a laptop CPU. We trained 60 different models with different random seeds to perform the experiments described in Section 5.3 and Figure 1. To calculate the LCE and apply LoRe, we used the final hidden layer representation learned by our model, applying t-SNE to reduce the dimension to 2 or PCA to reduce their dimension to 20, as a feature representation for the kernel.

A.3 CELEBA

For all experiments with the CelebA dataset, we trained a ResNet50 model and used it as our base classifier. We applied standard data augmentation to our training data (random crops & random horizontal flips), and trained all models for 10 epochs using the Adam optimizer with a learning rate of 1×10^{-3} and a batch size of 256. All other hyperparameters were PyTorch defaults. Training was distributed over 4 GPUs, and training a single model took about 30 minutes. For both Setting 2 and Setting 3 (described in Section 5.3), we trained 20 models with different random seeds to perform the experiments shown in Figures 2 and 3. To calculate the LCE and apply LoRe, we used pre-trained Inception-v3 features, applying t-SNE to reduce their dimension to 2 or PCA to reduce their dimension to 50, as a feature representation for the kernel.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

A.4 COMPAS CRIMINAL RECIDIVISM

For all experiments with the COMPAS criminal recidivism dataset, we used a 3-hidden-layer dense neural network as our base classifier. Each hidden layer had a width of 100 and was followed by a Leaky ReLU activation. We applied dropout with probability 0.4 after the final hidden layer. We trained the model using the Adam optimizer with a batch size of 64 and a learning rate of 3×10^{-4} until the validation accuracy stopped improving. All other hyperparameters were PyTorch defaults. Training was done locally on a laptop CPU. We trained 60 different models with different random seeds to perform the experiments described in Section 5.3 and Figure 1. To calculate the LCE and apply LoRe, we used the final hidden layer representation learned by our model, applying t-SNE to reduce the dimension to 2 or PCA to reduce their dimension to 20, as a feature representation for the kernel.

B ADDITIONAL EXPERIMENTAL RESULTS

In Figures 1, 2, 3, and 4 we visualize the MLCE achieved by all recalibration methods for the three experimental settings evaluated in Section 5.3. Figure 4 in the main paper shows the same visualization for all methods on ImageNet. In Figure 5, we plot the MLCE achieved by all recalibration methods for CIFAR-100, and in Figure 6, we do the same for CIFAR-10. Across all settings and datasets, our method LoRe is the most effective at minimizing MLCE across a wide range of γ, even accounting for variations between runs.

In these figures, "Original" represents no recalibration, "TS" represents temperature scaling, "HB" represents histogram binning, "IR" represents isotonic regression, "MMCE" represents direct MMCE optimization, and "LoRe" is our method.

Next, we examine the influence of the specific feature map used. In Figures 7, 8, 9, and 10, we plot the MLCE achieved by all recalibration methods for ImageNet using Inception-v3, AlexNet, DenseNet121, and ResNet101 features. In Figures 11 and 12, we plot the MLCE achieved by all recalibration methods for ImageNet when the features used to calculate the MLCE are different from the features used by LoRe. For completeness, in Figures 13, 14, 15, and 16, we also visualize the average LCE for all experimental settings. All plots show similar results: LoRe performs best over a wide range of γ.

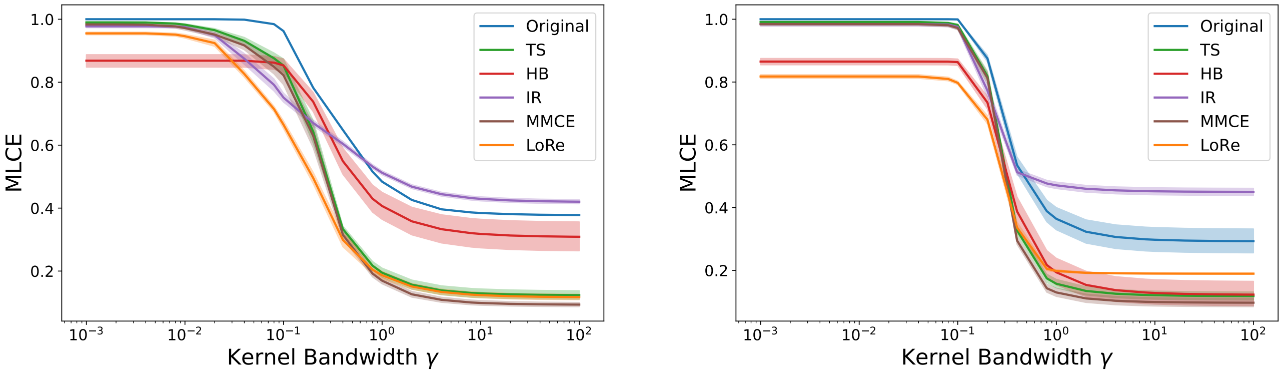

Figure 1: MLCE vs. kernel bandwidth γ for all methods on task 1 of Section 5.3, predicting whether a neighborhood's crime rate is higher than the median. LoRe achieves the best (or competitive) MLCE for most γ. Left: 2D t-SNE features. Right: 20D PCA features.

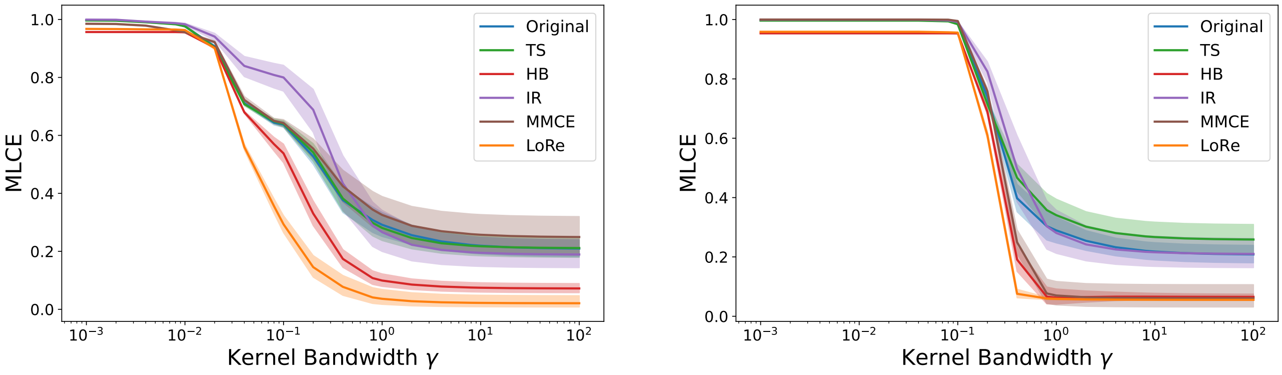

Figure 2: MLCE vs. kernel bandwidth γ for all methods on task 2 of Section 5.3, predicting hair color on CelebA. LoRe achieves the best MLCE for virtually all values of γ. Left: 2D t-SNE features. Right: 50D PCA features.

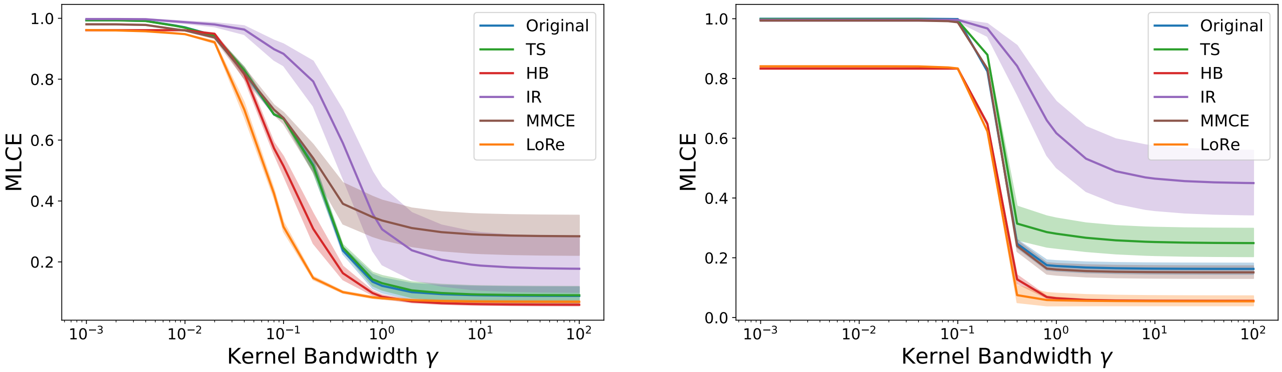

Figure 3: MLCE vs. kernel bandwidth for all methods on task 3 of Section 5.3, predicting hair type on CelebA. LoRe achieves the best MLCE for all $\gamma < 1$ and is tied with histogram binning for $\gamma > 1$. Left: 2D t-SNE features. Right: 50D PCA features.

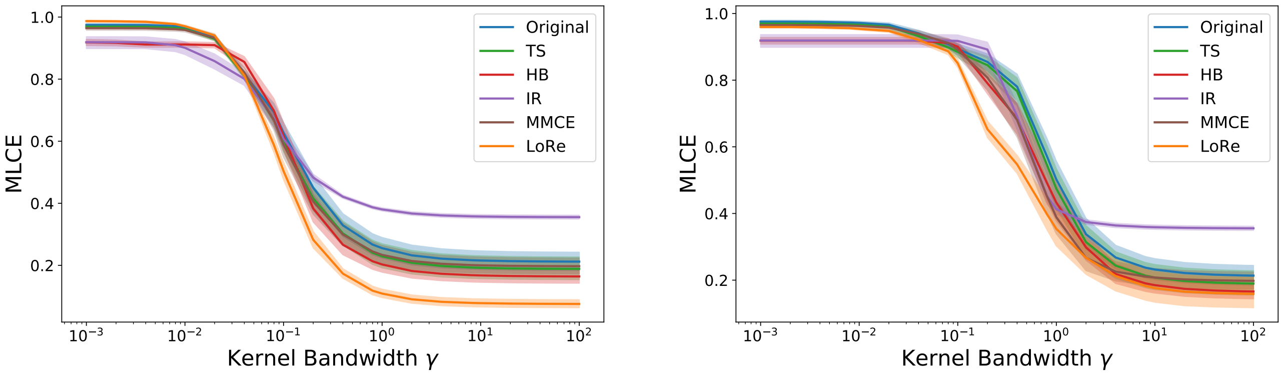

Figure 4: MLCE vs. kernel bandwidth for all methods on task 4 of Section 5.3, predicting criminal recidivism. LoRe achieves the best (or competitive) MLCE for most γ. Left: 2D t-SNE features. Right: 20D PCA features.

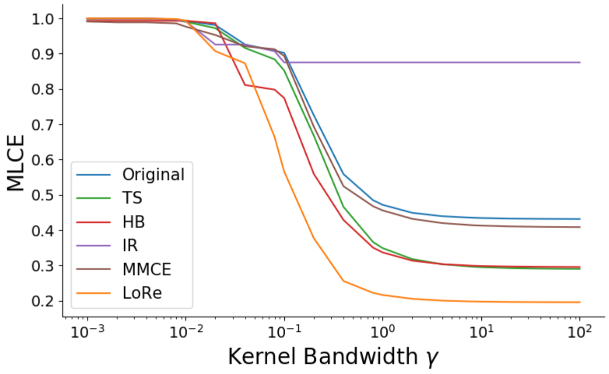

Figure 5: MLCE vs. kernel bandwidth γ for all recalibration methods for CIFAR-100 (3D t-SNE features). LoRe achieves lower MLCE for most γ.

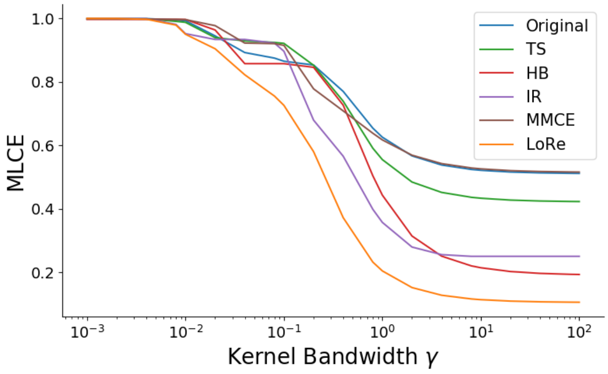

Figure 6: MLCE vs. kernel bandwidth γ for all recalibration methods for CIFAR-10 (3D t-SNE features). LoRe achieves lower MLCE for most γ.

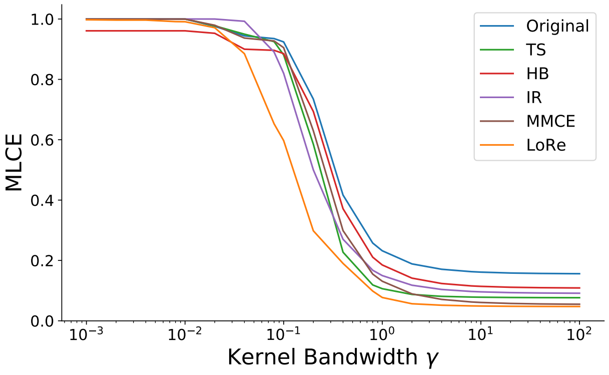

Figure 7: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using Inception-v3 features. LoRe achieves the best MLCE for most γ.

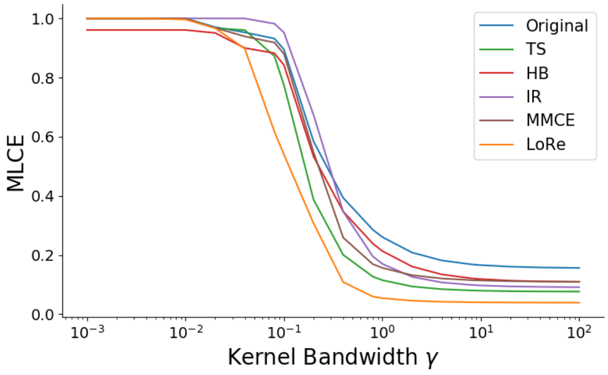

Figure 8: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using AlexNet features. LoRe achieves the best MLCE for most γ.

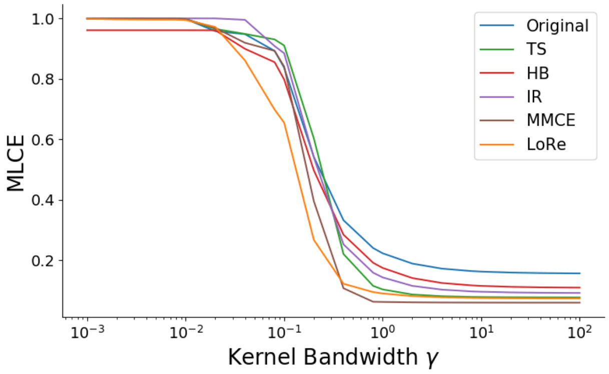

Figure 9: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using DenseNet121 features. LoRe achieves the best MLCE for most γ.

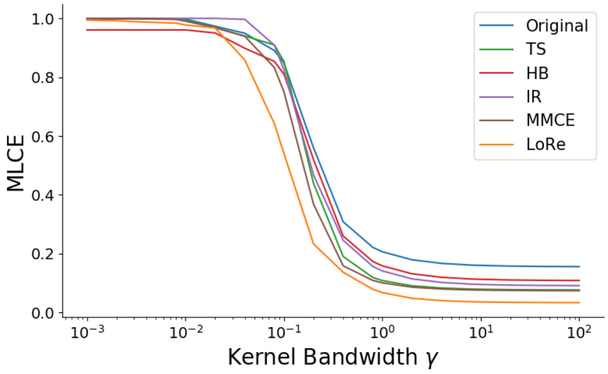

Figure 10: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using ResNet101 features. LoRe achieves the best MLCE for most γ.

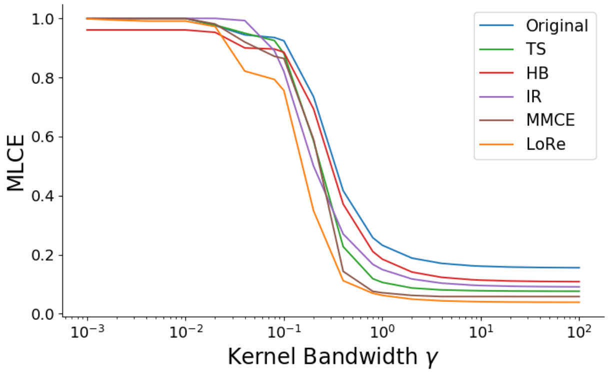

Figure 11: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using Inception-v3 features to calculate the MLCE and AlexNet features for applying LoRe.

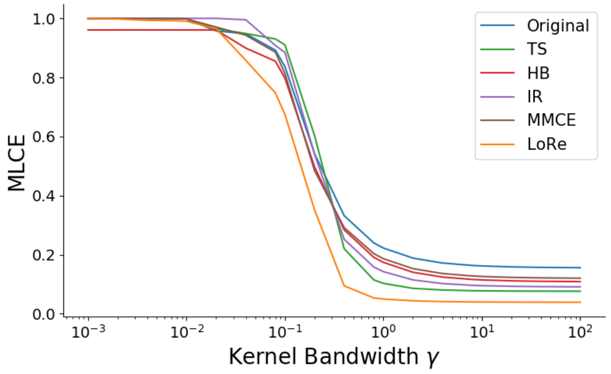

Figure 12: MLCE vs. kernel bandwidth γ for all recalibration methods on ImageNet using DenseNet121 features to calculate the MLCE and AlexNet features for applying LoRe.

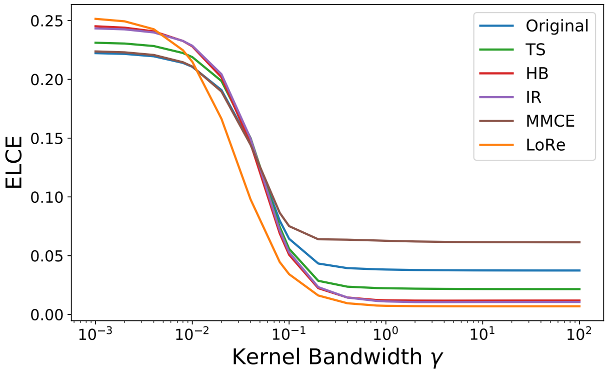

Figure 13: Average LCE vs. kernel bandwidth γ for all recalibration methods on ImageNet (3D t-SNE features). LoRe gets lower average LCE for most γ.

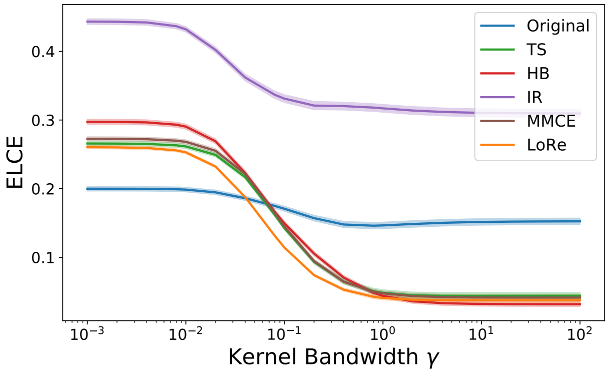

Figure 14: Average LCE vs. kernel bandwidth γ for all recalibration methods in task 1 (crime data, 2D t-SNE features). LoRe gets lower average LCE for most γ.

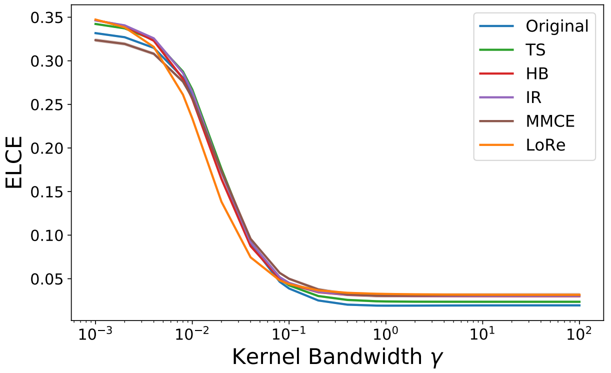

Figure 15: Average LCE vs. kernel bandwidth γ for all recalibration methods in task 2 (CelebA, 2D t-SNE features). LoRe gets lower average LCE for most γ.

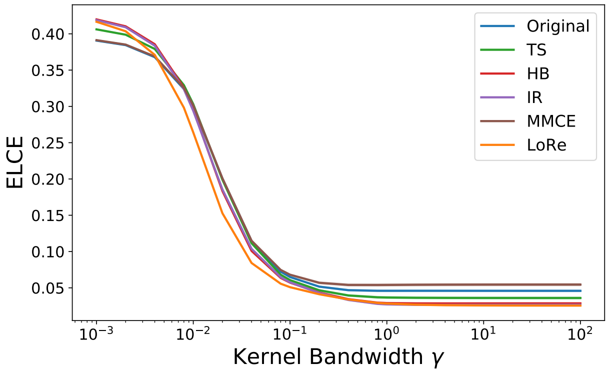

Figure 16: Average LCE vs. kernel bandwidth γ for all recalibration methods in task 3 (CelebA, 2D t-SNE features). LoRe gets lower average LCE for most γ.

C PROOF OF LEMMA 1

We restate Lemma 1 below, and provide the proof:

Lemma 1. *Assume that* $\lim_{\gamma\to\infty} k_\gamma(x, x') = 1$ *for all* $x, x' \in \mathcal{X}$*. Then, as* $\gamma \to \infty$*, the MLCE converges to the MCE.*

Proof. Since $\lim_{\gamma\to\infty} k_\gamma(x, x') = 1$ identically,

$$\begin{aligned}
\lim_{\gamma\to\infty} \max_x \widehat{\mathrm{LCE}}_\gamma(x; f, \hat{p}) &= \max_x \frac{1}{|\beta(x)|}\left|\sum_{i\in\beta(x)} \hat{p}(x_i) - \mathbb{1}\left[f(x_i) = y_i\right]\right| \\
&= \max_k \frac{1}{|B_k|}\left|\sum_{i\in B_k} \hat{p}(x_i) - \mathbb{1}\left[f(x_i) = y_i\right]\right| \\
&= \max_k |\mathrm{conf}(B_k) - \mathrm{acc}(B_k)| \\
&= \mathrm{MCE}(x; f, \hat{p})
\end{aligned}$$

□

D FORMAL STATEMENT AND PROOF OF THEOREM 1

Let $B_1, \dots, B_N$ denote a set of bins that partition $[0, 1]$, and $B(p)$ denote the bin that a particular $p \in [0, 1]$ belongs to. Let $a_f(x, y) = \mathbb{1}\left[f(x) = y\right]$ indicate the accuracy of a the classifier $(f, \hat{p})$ on an input x. We consider the signed local calibration error (SLCE):

$$\begin{aligned}
\mathrm{SLCE}_\gamma(x; f, \hat{p}) &:= \frac{\mathbb{E}[(\hat{p}(X) - a_f(X, Y))k_\gamma(X, x) \mid \hat{p}(X) \in B(\hat{p}(x))]}{\mathbb{E}[k_\gamma(X, x) \mid \hat{p}(X) \in B(\hat{p}(x))]} \\
&= \frac{\mathbb{E}[(\hat{p}(X) - a_f(X, Y))k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]]}{\mathbb{E}[k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]]}.
\end{aligned}$$

D.1 ASSUMPTIONS AND FORMAL STATEMENT OF THEOREM

We make the following assumptions:

Assumption A (Lipschitz kernel)**.** *The kernel* k_γ *takes the form*

$$k_\gamma(x, x') = g\left(\frac{\phi(x) - \phi(x')}{\gamma}\right),$$

where $\phi : \mathcal{X} \to \mathbb{R}^d$ *is a representation function, and* $g : \mathbb{R}^d \to [0, 1]$ *is L-Lipschitz with respect to some norm* $\|\cdot\|$*.*

Note this definition may require an implicit rescaling (for example, we can take $\phi(x) \leftarrow \phi^{\mathrm{feature}}(x)/d$ for a d-dimensional feature map ϕ^{feature} and take $g(z) = \exp(-\|z\|_1)$, which corresponds to the Laplacian kernel we used in Section 3.2).

Assumption B (Binning-aware covering number)**.** *For any* $\epsilon > 0$*, the range of the representation function* $\phi(\mathcal{X}) := \{\phi(x) : x \in \mathcal{X}\}$ *has an* ϵ*-cover in the* $\|\cdot\|$*-norm of size* $(C/\epsilon)^d$ *for some absolute constant* $C > 0$*: There exists a set* $\mathcal{N}_\epsilon \in \mathcal{X}$ *with* $|\mathcal{N}_\epsilon| \le (C/\epsilon)^d$ *such that for any* $x \in \mathcal{X}$*, there exists some* $x' \in \mathcal{N}_\epsilon$ *such that* $\|\phi(x) - \phi(x')\| \le \epsilon$ *and* $B(\hat{p}(x)) = B(\hat{p}(x'))$*.*

Assumption C (Lower bound on expectation of kernel within bin)**.** *We have*

$$\inf_{x\in\mathcal{X}} \mathbb{E}[k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]] \ge \alpha$$

for some constant $\alpha \in (0, 1)$*.*

The constant α characterizes the hardness of estimating the SLCE from samples. Intuitively, with a smaller α, the denominator in SLCE gets smaller and we desire a higher accuracy in estimating both the numerator and the denominator. Also note that in practice the value of α typically depends on γ.

We analyze the following estimator of the SLCE using n samples:

$$\widehat{\mathrm{SLCE}}_\gamma(x; f, \hat{p}) = \frac{\frac{1}{n}\sum_{i=1}^n(\hat{p}(x_i) - a_f(x_i, y_i))k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]}{\frac{1}{n}\sum_{i=1}^n k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]}. \tag{1}$$

Theorem 1. *Under Assumptions A, B, and C, Suppose the sample size $n \geq \widetilde{O}(d/\alpha^4\epsilon^2)$ where $\epsilon > 0$ is a target accuracy level, then with probability at least $1-\delta$ we have*

$$\sup_{x\in\mathcal{X}} \left|\widehat{\mathrm{SLCE}}_\gamma(x; f, \hat{p}) - \mathrm{SLCE}_\gamma(x; f, \hat{p})\right| \leq \epsilon,$$

where $\widetilde{O}$ hides log factors of the form $\log(L/\gamma\epsilon\delta\alpha)$.

Theorem 1 shows that $\widetilde{O}(d/\epsilon^2\alpha^4)$ samples is sufficient to estimate the SLCE simultaneously for all $x \in \mathcal{X}$. When $\alpha = \Omega(1)$, this sample complexity only depends polynomially in terms of the representation dimension d and logarithmically in other constants (such as L, γ, and the failure probability δ).

D.2 PROOF OF THEOREM 1

Step 1. We first study the estimation at finitely many x's. Let $\mathcal{N} \subseteq \mathcal{X}$ be a finite set of x's with $|\mathcal{N}| = N$. Since $k_\gamma \in [0, 1]$ and $|\hat{p}(x) - a_f(x, y)| \leq 1$ are bounded variables, by the Hoeffding inequality and a union bound, we have

$$\begin{aligned}
&\mathbb{P}\Bigg(\sup_{x\in\mathcal{N}}\Bigg|\frac{1}{n}\sum_{i=1}^n(\hat{p}(x_i) - a_f(x_i, y_i))k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right] \\
&\qquad - \mathbb{E}[(\hat{p}(X) - a_f(X, Y))k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]]\Bigg| > \alpha\epsilon/10\Bigg) \\
&\leq \exp\big(-cn\alpha^2\epsilon^2 + \log N\big).
\end{aligned}$$

Therefore, as long as $n \geq O(\log(N/\delta)/\epsilon^2\alpha^2)$ samples, the above probability is bounded by δ. In other words, with probability at least $1 - \delta$, we have simultaneously

$$\begin{aligned}
&\Bigg|\underbrace{\frac{1}{n}\sum_{i=1}^n(\hat{p}(x_i) - a_f(x_i, y_i))k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]}_{:=\hat{A}(x)} \\
&\qquad - \underbrace{\mathbb{E}[(\hat{p}(X) - a_f(X, Y))k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]]}_{:=A(x)}\Bigg| \\
&\leq \alpha\epsilon/10.
\end{aligned}$$

for all $x \in \mathcal{N}$. Similarly, when $n \geq O(\log(N/\delta)/\epsilon^2\alpha^4)$, we also have (with probability at least $1 - \delta$)

$$\Bigg|\underbrace{\frac{1}{n}\sum_{i=1}^n k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]}_{:=\hat{B}(x)} - \underbrace{\mathbb{E}[k_\gamma(X, x)\mathbb{1}\left[\hat{p}(X) \in B(\hat{p}(x))\right]]}_{:=B(x)}\Bigg| \leq \alpha^2\epsilon/10$$

On these concentration events, we have for any $x \in \mathcal{N}$ that

$$\begin{aligned}
\left|\widehat{\mathrm{SLCE}}_\gamma(x; f, \hat{p}) - \mathrm{SLCE}_\gamma(x; f, \hat{p})\right| &= \left|\frac{\hat{A}(x)}{\hat{B}(x)} - \frac{A(x)}{B(x)}\right| \\
&\leq \left|\hat{A}(x)\right|\left|\frac{1}{\hat{B}(x)} - \frac{1}{B(x)}\right| + \frac{1}{|B(x)|}\left|\hat{A}(x) - A(x)\right| \\
&\leq 1 \cdot \frac{\alpha^2\epsilon/10}{\alpha(\alpha - \alpha^2\epsilon/10)} + \frac{1}{\alpha}\cdot\alpha\epsilon/10 \\
&\leq \epsilon.
\end{aligned}$$

Step 2. We now extend the bound to all $x \in \mathcal{X}$ using the covering argument. By Assumption B, we can take an $\alpha^2\epsilon\gamma/(10L)$-covering of $\phi(\mathcal{X})$ with cardinality $N \leq (10CL/\alpha^2\epsilon\gamma)^d$. Let $\mathcal{N} \subset \mathcal{X}$ denote the covering set (in the $\mathcal{X}$ space). This means that for any $x \in \mathcal{X}$, there exists $x' \in \mathcal{N}$ such that $\|\phi(x) - \phi(x')\| \leq \alpha^2\epsilon\gamma/(10L)$ amd $B(\hat{p}(x)) = B(\hat{p}(x'))$, which implies that for any $\widetilde{x} \in \mathcal{X}$ we have

$$\begin{aligned}
|k(\widetilde{x}, x) - k(\widetilde{x}, x')| &= \left| f\left(\frac{\phi(\widetilde{x}) - \phi(x)}{\gamma}\right) - f\left(\frac{\phi(\widetilde{x}) - \phi(x')}{\gamma}\right)\right| \\
&\leq \frac{L}{\gamma}\, \|\phi(x) - \phi(x')\| \\
&\leq \alpha^2\epsilon/10,
\end{aligned}$$

where we have used the Lipschitzness assumption of g (Assumption A). This further implies

$$\begin{aligned}
\left|\hat{A}(x) - \hat{A}(x')\right| &= \left|\frac{1}{n}\sum_{i=1}^{n}(\hat{p}(x_i) - a_f(x_i, y_i))k_\gamma(x_i, x)\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]\right. \\
&\qquad \left. - \frac{1}{n}\sum_{i=1}^{n}(\hat{p}(x_i) - a_f(x_i, y_i))k_\gamma(x_i, x')\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x'))\right]\right| \\
&= \left|\frac{1}{n}\sum_{i=1}^{n}(\hat{p}(x_i) - a_f(x_i, y_i))[k_\gamma(x_i, x) - k_\gamma(x_i, x')]\mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right]\right| \\
&\leq \frac{1}{n}\sum_{i=1}^{n}|\hat{p}(x_i) - a_f(x_i, y_i)| \cdot |k_\gamma(x_i, x) - k_\gamma(x_i, x')| \cdot \mathbb{1}\left[\hat{p}(x_i) \in B(\hat{p}(x))\right] \\
&\leq \alpha^2\epsilon/10.
\end{aligned}$$

Similarly, we have $|A(x) - A(x')| \leq \alpha^2\epsilon/10$, $|\hat{B}(x) - \hat{B}(x')| \leq \alpha^2\epsilon/10$, and $|B(x) - B(x')| \leq \alpha^2\epsilon/10$. This means that the estimation error at x is close to that at $x' \in \mathcal{N}$ and consequently also bounded by ϵ:

$$\begin{aligned}
\left|\widehat{\mathrm{SLCE}}_\gamma(x; f, \hat{p}) - \mathrm{SLCE}_\gamma(x; f, \hat{p})\right| &= \left|\frac{\hat{A}(x)}{\hat{B}(x)} - \frac{A(x)}{B(x)}\right| \\
&\leq \left|\frac{\hat{A}(x)}{\hat{B}(x)} - \frac{\hat{A}(x')}{\hat{B}(x')}\right| + \left|\frac{\hat{A}(x')}{\hat{B}(x')} - \frac{A(x')}{B(x')}\right| + \left|\frac{A(x')}{B(x')} - \frac{A(x)}{B(x)}\right| \\
&\leq 3\left[1 \cdot \frac{\alpha^2\epsilon/10}{\alpha(\alpha - \alpha^2\epsilon/10)} + \frac{1}{\alpha} \cdot \alpha^2\epsilon/10\right] \\
&\leq \epsilon.
\end{aligned}$$

Therefore, taking this $\mathcal{N}$ in step 1, we know that as long as the sample size

$$N \geq O\left(\frac{\log(|\mathcal{N}|/\delta)}{\epsilon^2\alpha^4}\right) = O\left(\frac{d\big[\log\big(10CL/\alpha^2\epsilon\gamma\big) + \log(1/\delta)\big]}{\alpha^4\epsilon^2}\right) = \widetilde{O}\big(d/\alpha^4\epsilon^2\big),$$

we have with probability at least $1 - \delta$ that

$$\sup_{x \in \mathcal{X}} \left|\widehat{\mathrm{SLCE}}_\gamma(x; f, \hat{p}) - \mathrm{SLCE}_\gamma(x; f, \hat{p})\right| \leq \epsilon.$$

This is the desired result.

□