Appendix: Active Learning with Maximum Margin Sparse Gaussian Processes

Organization. The appendix provides important details that help to understand the key theoretical results and major technical components presented in the main paper. It also contains additional experimental results on the real-world data that complements the result presented in the main paper. The organization is as follows: Appendix A proves the major theoretical results, including Lemma 1 and Theorem 1, presented in Sect. 3. Appendix B presents the details of the hyperparameter learning results. Appendix C describes the construction of the augmentation set that is used to conduct the one-time joint augmentation. Appendix 4.1 shows the experimental results from the synthetic data that help to demonstrate important properties of the proposed MM-SGP model and the sampling function. Finally, Appendix D shows some additional results on the real-world datasets.

A Proof of Theoretical Results

In this section, we provide the detailed proof of the major theoretical results in the paper.

A.1 Proof of Lemma 1

Proof We start by formulating the Lagrangian function of (1),

$$\begin{aligned} L(q(\mathbf{w}), \boldsymbol{\xi}, \overline{\mathbf{X}}, \boldsymbol{\gamma}) = c\sum_n \xi_n + KL(q(\mathbf{w})||p(\mathbf{w}|\overline{\mathbf{X}})) - \mathbb{E}_{q(\mathbf{w})}[\log g(\mathbf{w}, \boldsymbol{\gamma}, \overline{\mathbf{X}})] \\ - \sum_n \alpha_n \left\{\mathbb{E}_{q(\mathbf{w})}[y_n(\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n})\right\} - \Delta l_n(y) + \xi_n) + \alpha_0 \left[\int q(\mathbf{w})d\mathbf{w} - 1\right] \end{aligned} \tag{15}$$

We proceed by taking the partial derivative of L over $q(\mathbf{w})$. Using the calculus of variations, we get:

$$\frac{\partial L}{\partial q(\mathbf{w})} = \ln q(\mathbf{w}) - \ln p(\mathbf{w}|\overline{\boldsymbol{X}}) + \alpha_0 - \ln g(\mathbf{w}, \boldsymbol{\gamma}, \overline{X}) - \sum_{n=1}^{N} \alpha_n y_n(\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n}) \tag{16}$$

We then substitute the local variational bound $g(\mathbf{w}, \boldsymbol{\gamma}, \overline{\mathbf{X}})$ as defined by (2) and the SGP prior $p(\mathbf{w}|\overline{\mathbf{X}}) \sim \mathcal{N}(0, K_{MM}^{-1})$ into (16). By setting it to zero, we have

$$\begin{aligned} \ln q(\mathbf{w}) =& -\frac{M}{2}\ln(2\pi) + \frac{1}{2}\ln|K_{MM}^{-1}| - \frac{1}{2}\mathbf{w}^T K_{MM}^{-1}\mathbf{w} - \alpha_0 \\ &+ \sum_{n=1}^{N}\Big\{\ln\sigma(\gamma_n) + \lambda(\gamma_n)\gamma_n^2 - \frac{1}{2}\gamma_n + (y_n - \frac{1}{2})\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n} \\ &-\lambda(\gamma_n)\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n}\mathbf{k}_{\mathbf{x}_n}^T\mathbf{w} + \alpha_n y_n \mathbf{w}^T\mathbf{k}_{\mathbf{x}_n}\Big\} \\ =& -\frac{1}{2}\mathbf{w}^T\left[K_{MM}^{-1} + 2\sum_{n=1}^{N}\lambda(\gamma_n)\mathbf{k}_{\mathbf{x}_n}\mathbf{k}_{\mathbf{x}_n}^T\right]\mathbf{w} + \sum_{n=1}^{N}\left[(\alpha_n + \frac{1}{2})y_n\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n}\right] + \text{const} \end{aligned} \tag{17}$$

where terms irrelevant to $\mathbf{w}$ are absorbed into the const term. By identifying the linear and quadratic terms of $\mathbf{w}$, we complete the square to get

$$q(\mathbf{w}) \sim \mathcal{N}(\mu_q(\boldsymbol{\alpha}), \Sigma_q^{-1}) \tag{18}$$

$$\boldsymbol{\mu}_q(\boldsymbol{\alpha}) = \Sigma_q\left[\sum_{n=1}^{N}(\alpha_n + \frac{1}{2})y_n\mathbf{k}_{\mathbf{x}_n}\right] \tag{19}$$

$$\Sigma_q^{-1} = K_{MM} + 2\sum_{n=1}^{N}\lambda(\gamma_n)\mathbf{k}_{\mathbf{x}_n}\mathbf{k}_{\mathbf{x}_n}^T \tag{20}$$

A.2 Proof of Theorem 1

Proof We start by showing the dual problem of (1) that solves the Lagrangian multipliers. To proceed, we first substitute $q(\mathbf{w})$ back to L and with some algebra, we arrive at a general formulation similar to [6]:

$$\max_{\boldsymbol{\alpha}} -\ln Z(\boldsymbol{\alpha}) - U^*(\boldsymbol{\alpha}) \quad \text{s.t.} \quad \alpha_i \geq 0 \tag{21}$$

where $Z(\boldsymbol{\alpha})$ is the normalization term of $q(\mathbf{w})$ and $U^*(\boldsymbol{\alpha})$ is the conjugate of the slack function given by $U^*(\boldsymbol{\alpha}) = \sup(\sum_{n=1}^N \alpha_n \xi_n - c\sum_{n=1}^N \xi_n)$. Here, we leverage the property that the conjugate of the convex function $U(\boldsymbol{\xi})$, which is given by $U(\boldsymbol{\alpha})^* = \sup[\boldsymbol{\alpha}^T\boldsymbol{\xi} - U(\boldsymbol{\xi})]$ in the maximum margin setting [29]. Substitute the specific form of the slack function, $U(\boldsymbol{\xi}) = c\sum_{n=1}^N \xi_n$, it can be shown that

$$U^*(\boldsymbol{\alpha}) = \begin{cases} 0 & \sum_{n=1}^N \alpha_n \leq c \\ \infty & \text{otherwise} \end{cases}$$

Furthermore, the inequality can be ignored since its corresponding solution, $\boldsymbol{\xi} = 0$ is still included. Therefore, we use constraints $\sum_{n=1}^N \alpha_n = c$ to replace $-U^*(\boldsymbol{\alpha})$ in (21).

We further compute $Z(\boldsymbol{\alpha})$ by integrating out $\mathbf{w}$:

$$\begin{aligned} Z(\boldsymbol{\alpha}) = &\int (2\pi)^{-\frac{M}{2}} |K_{MM}|^{\frac{1}{2}} \prod_{n=1}^N \sigma(\gamma_n) \exp\left\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu}_q(\boldsymbol{\alpha}))^T \Sigma_q^{-1}(\mathbf{w} - \boldsymbol{\mu}_q(\boldsymbol{\alpha}))\right. \\ &\left. + \frac{1}{2}\boldsymbol{\mu}_q(\boldsymbol{\alpha})^T\Sigma_q^{-1}\boldsymbol{\mu}_q(\boldsymbol{\alpha}) + \sum_{n=1}^N \lambda(\gamma_n)\gamma_n^2 - \frac{1}{2}\gamma_n\right\} d\mathbf{w} \qquad (22) \\ = &\left(\frac{|K_{MM}|}{|\Sigma_q|}\right)^{\frac{1}{2}} \prod_{n=1}^N \sigma(\gamma_n) \exp\left\{\frac{1}{2}\boldsymbol{\mu}^{*T}\Sigma_q\boldsymbol{\mu}^* + \sum_{n=1}^N \lambda(\gamma_n)\gamma_n^2 - \frac{1}{2}\gamma_n\right\} \qquad (23) \end{aligned}$$

where $\boldsymbol{\mu}^* = \sum_{n=1}^N (\alpha_n + \frac{1}{2}) y_n \mathbf{k}_{\mathbf{x}_n}$ Substituting $Z(\boldsymbol{\alpha})$ back to (21), and removing terms irrelevant to $\boldsymbol{\alpha}$, we arrive at the dual problem given by (6), where

$$Q = \Lambda K_{NM}\Sigma_q K_{NM}^T\Lambda, \quad \Lambda = \text{diag}(\mathbf{y}) \tag{24}$$

Since Σ_q is positive definite (according to (20)), we have $\Sigma_q = (\Sigma_q^{\frac{1}{2}})^2$. Therefore, we can represent Q as $Q = VV^T$, where $V = \Lambda K_{NM}\Sigma_q^{\frac{1}{2}}$. Since the rank of matrix V is at most M, the bound of time complexity of each step in the iterative QP solving process is reduced to $O(NM^2)$ from $O(N^3)$.

B Details of Hyperparameter Learning

In this section, we provide details of learning the key hyperparameters in the model.

B.1 Learning Variational Local Parameter $\boldsymbol{\gamma}$

To derive the closed-form update rule for $\boldsymbol{\gamma}$, we set $\frac{\partial L}{\partial \boldsymbol{\gamma}} = 0$

$$\begin{aligned} \frac{\partial L}{\partial \gamma_n} =& \frac{\partial}{\partial \gamma_n}\mathbb{E}\left[\sum_{n=1}^N \ln\sigma(\gamma_n) - \frac{\gamma_n}{2} - \lambda(\gamma_n)(\mathbf{k}_{\mathbf{x}_n}^T\mathbf{w}\mathbf{w}^T\mathbf{k}_{\mathbf{x}_n} - \gamma_n^2)\right] \qquad (25) \\ =& \lambda'(\gamma_n)(\mathbf{k}_{\mathbf{x}_n}^T\mathbb{E}[\mathbf{w}\mathbf{w}^T]\mathbf{k}_{\mathbf{x}_n} - \gamma_n^2) \qquad (26) \end{aligned}$$

where we have used $\mathrm{d}\sigma = \sigma(1-\sigma)$. From the definition of $\lambda(\gamma_n)$, it is easy to see that $\lambda'(\gamma_n)$ is a monotonic function for $\gamma_n \geq 0$. Since γ_n is non-negative, we have $\lambda'(\gamma_n) \neq 0$ and solving (26) leads to

$$\gamma_n^2 = \mathbf{k}_{\mathbf{x}_n}^T\mathbb{E}[\mathbf{w}\mathbf{w}^T]\mathbf{k}_{\mathbf{x}_n} = \mathbf{k}_{\mathbf{x}_n}^T(\Sigma_q + \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T)\mathbf{k}_{\mathbf{x}_n} \tag{27}$$

B.2 Learning the Support Set

A key step to update $\overline{\mathbf{X}}$ is to express the Lagrangian function (15) as the function of $\overline{\mathbf{X}}$. After that, we can resort to standard gradient based optimization method to solve for $\overline{\mathbf{X}}$. First, we represent both K_{NM} and K_{MM} as functions of $\overline{\mathbf{X}}$: $K_{NM} = F(\overline{\mathbf{X}}), K_{MM} = G(\overline{\mathbf{X}})$. We can reformulate the moments of $\mathbf{w}$ as:

$$\mu_q(\boldsymbol{\alpha}) = \Sigma_q F(\overline{\mathbf{X}})^T \Lambda \widehat{\boldsymbol{\alpha}} \tag{28}$$

$$\Sigma_q^{-1} = G(\overline{\mathbf{X}}) + F(\overline{\mathbf{X}})^T \Gamma F(\overline{\mathbf{X}}) \tag{29}$$

Then we express L as a function of F, G and the moments of $\mathbf{w}$. Specifically, there are three terms in L related to $\overline{\mathbf{X}}$. The first term is the KL divergence:

$$KL(q(\mathbf{w})||p(\mathbf{w}|\overline{\mathbf{X}})) = \frac{1}{2}\left(\ln|G(\overline{\mathbf{X}})| - \ln|\Sigma_q| + \mathrm{Tr}[G(\overline{\mathbf{X}})^{-1}\Sigma_q] + \mu_q(\boldsymbol{\alpha})^T G(\overline{\mathbf{X}})^{-1}\mu_q(\boldsymbol{\alpha})\right) \tag{30}$$

The second term is the expected log of the variational lower bound (approximation of the likelihood):

$$\mathbb{E}_{q(\mathbf{w})}[\log g(\mathbf{w}, \boldsymbol{\gamma}, \overline{\mathbf{X}})] = \mathbf{y}^T F(\overline{\mathbf{X}})\mu_q(\boldsymbol{\alpha}) - \frac{1}{2}\mathbf{1}^T F(\overline{\mathbf{X}})\mu_q(\boldsymbol{\alpha}) - \mathrm{Tr}[F(\overline{\mathbf{X}})(\Sigma_q + \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T)F(\overline{\mathbf{X}})^T] \tag{31}$$

The last term is from the expected max-margin constraint:

$$\sum_n \alpha_n \left\{\mathbb{E}_{q(\mathbf{w})}[y_n(\mathbf{w}^T \mathbf{k}_{\mathbf{x}_n})\right\} = \boldsymbol{\alpha}^T \Lambda F(\overline{\mathbf{X}})\mu_q(\boldsymbol{\alpha}) \tag{32}$$

For any differentiable kernel function, we first compute the gradients $\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}$ and $\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}$. Then we substitute them back to (30), (31), and (32) and apply the chain rule to get the $\frac{\partial L}{\partial \overline{\mathbf{X}}}$ and solve for $\overline{\mathbf{X}}$ with gradient decent.

To simplify the expression, we first define the following terms:

- **Term 1**:

$$\frac{\partial \Sigma_q}{\partial \overline{\mathbf{X}}} = \Sigma_q\left(\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}} + \frac{2\partial F(\overline{\mathbf{X}})^T \Gamma F(\overline{\mathbf{X}})}{\partial F(\overline{\mathbf{X}})}\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\right)\Sigma_q \tag{33}$$

  where

$$\frac{\partial (F(\overline{\mathbf{X}})^T \Gamma F(\overline{\mathbf{X}}))}{\partial F(\overline{\mathbf{X}})_{ij}} = F(\overline{\mathbf{X}})^T \Gamma J^{ij} + J^{ji}\Gamma F(\overline{\mathbf{X}}) \tag{34}$$

  J^{ij} is a single - entry matrix with 1 at its (i, j)-th element and zeros elsewhere. We also have $J^{ij}_{kl} = \delta_{ik}\delta_{jl}$ in addition where k, l are indices of the first and second dimensions of $F(\overline{\mathbf{X}})^T \Gamma F(\overline{\mathbf{X}})$. It might be easier to understand (34) element-wisely:

$$\frac{\partial (F(\overline{\mathbf{X}})^T \Gamma F(\overline{\mathbf{X}}))_{kl}}{\partial F(\overline{\mathbf{X}})_{ij}} = \delta_{lj}(F(\overline{\mathbf{X}})^T \Gamma)_{ki} + \delta_{kj}(\Gamma F(\overline{\mathbf{X}}))_{il} \tag{35}$$

- **Term 2**:

$$\frac{\partial \mu_q(\boldsymbol{\alpha})}{\partial \overline{\mathbf{X}}} = \left(\frac{\partial \Sigma_q}{\partial \overline{\mathbf{X}}} F(\overline{\mathbf{X}})^T + \Sigma_q\left(\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\right)^T\right)\Lambda \widehat{\boldsymbol{\alpha}} \tag{36}$$

- **Term 3**: (We focus on the two kernel functions used by our experiments and other differentiable kernels can be similarly derived.)

  When a *linear kernel* is applied ($F(\overline{\mathbf{X}})_{kl} = \mathbf{x}_k^T \overline{\mathbf{x}}_l$),

$$\frac{\partial F(\overline{\mathbf{X}})_{kl}}{\partial \overline{\mathbf{x}}_j} = \delta_{jl}\mathbf{x}_k^T \tag{37}$$

  When a *RBF kernel* is applied ($F(\overline{\mathbf{X}})_{kl} = \exp\left(-\frac{(\mathbf{x}_k - \overline{\mathbf{x}}_l)^T(\mathbf{x}_k - \overline{\mathbf{x}}_l)}{2\sigma^2}\right)$),

$$\frac{\partial F(\overline{\mathbf{X}})_{kl}}{\partial \overline{\mathbf{x}}_j} = \exp\left(-\frac{(\mathbf{x}_k - \overline{\mathbf{x}}_l)^T(\mathbf{x}_k - \overline{\mathbf{x}}_l)}{2\sigma^2}\right)\left(\frac{\delta_{lj}(\hat{\mathbf{x}}_l - 2\mathbf{x}_k)}{-2\sigma^2}\right) \tag{38}$$

  where $\overline{\mathbf{x}}_j$ denotes the j-th row of $\overline{\mathbf{X}}$

- **Term 4**:

  When linear kernel is applied ($G(\overline{\mathbf{X}})_{kl} = \overline{\mathbf{x}}_k^T \overline{\mathbf{x}}_l$),

$$\frac{\partial G(\overline{\mathbf{X}})_{kl}}{\partial \overline{\mathbf{x}}_j} = \delta_{jl}\overline{\mathbf{x}}_k + \delta_{jk}\overline{\mathbf{x}}_l \tag{39}$$

  When RBF kernel is applied ($G(\overline{\mathbf{X}})_{kl} = \exp\left(-\frac{(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)^T(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)}{2\sigma^2}\right)$),

$$\frac{\partial G(\overline{\mathbf{X}})_{kl}}{\partial \overline{\mathbf{x}}_j} = \exp\left(-\frac{(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)^T(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)}{2\sigma^2}\right)\left(-\frac{\delta_{kj}\overline{\mathbf{x}}_k - 2(\delta_{kj}\overline{\mathbf{x}}_l + \delta_{lj}\overline{\mathbf{x}}_k) + \delta_{lj}\overline{\mathbf{x}}_l}{2\sigma^2}\right) \tag{40}$$

Now we can express the gradients of the objective as a function of terms 1-4. Specifically,

$$\begin{aligned}\frac{\partial KL(q(\mathbf{w})||p(\mathbf{w}|\overline{\mathbf{X}}))}{\partial \overline{\mathbf{X}}} &= \frac{1}{2}\left(\mathrm{Tr}[G(\overline{\mathbf{X}})^{-1}\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}] - \mathrm{Tr}[\Sigma_q^{-1}\frac{\partial \Sigma_q}{\partial \overline{\mathbf{X}}}]\right.\\ &+ G(\overline{\mathbf{X}})^{-1}\left(\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}G(\overline{\mathbf{X}})^{-1}\Sigma_q + \frac{\partial \Sigma_q}{\partial \overline{\mathbf{X}}}\right)\\ &+ \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T G(\overline{\mathbf{X}})^{-1}\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}G(\overline{\mathbf{X}})^{-1}\\ &+ \left.(G(\overline{\mathbf{X}})^{-1} + G(\overline{\mathbf{X}})^{-T})\mu_q(\boldsymbol{\alpha})\frac{\partial G(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\right)\end{aligned} \tag{41}$$

$$\begin{aligned}\frac{\partial \mathbb{E}_{q(\mathbf{w})}[\log g(\mathbf{w}, \boldsymbol{\gamma}, \overline{\mathbf{X}})]}{\partial \overline{\mathbf{X}}} &= (\mathbf{y}^T - \frac{1}{2}\mathbf{1}^T)\left(\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\mu_q(\boldsymbol{\alpha}) + F(\overline{\mathbf{X}})\frac{\partial \mu_q(\boldsymbol{\alpha})}{\partial \overline{\mathbf{X}}}\right)\\ &- F(\overline{\mathbf{X}})\left((\Sigma_q + \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T\right)\\ &- \left.(\Sigma_q + \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T)^T\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\right)\\ &+ F(\overline{\mathbf{X}})^T F(\overline{\mathbf{X}})\left(\frac{\partial \Sigma_q + \mu_q(\boldsymbol{\alpha})\mu_q(\boldsymbol{\alpha})^T}{\partial \overline{\mathbf{X}}}\right)\end{aligned} \tag{42}$$

$$\frac{\partial \boldsymbol{\alpha}^T \Lambda F(\overline{\mathbf{X}})\mu_q(\boldsymbol{\alpha})}{\partial \overline{\mathbf{X}}} = \boldsymbol{\alpha}^T\Lambda\left(\frac{\partial F(\overline{\mathbf{X}})}{\partial \overline{\mathbf{X}}}\mu_q(\boldsymbol{\alpha}) + F(\overline{\mathbf{X}})\frac{\partial \mu_q(\boldsymbol{\alpha})}{\partial \overline{\mathbf{X}}}\right) \tag{43}$$

The final gradient $\frac{\partial L}{\partial \overline{\mathbf{X}}}$ is given by the sum of (41), (42), (43) with terms 1-4 plugged in.

B.3 Learning Kernel Hyperparameters

In this section, we present the approach to update σ^2, the hyper-parameter of the RBF kernel. The dot product kernel used in our work does not have any tunable hyper-parameter. The derivation of the gradient is similar to section B.2. We only need to update the four terms as follow:

- **Term 1**:

$$\frac{\partial \Sigma_q}{\partial \sigma^2} = \frac{\partial G(\overline{\mathbf{X}})}{\partial \sigma^2} + 2F(\overline{\mathbf{X}})(\Gamma + \Gamma^T)\frac{\partial F(\overline{\mathbf{X}})}{\partial \sigma^2} \tag{44}$$

- **Term 2**:

$$\frac{\partial \mu_q(\boldsymbol{\alpha})}{\partial \sigma^2} = \left(\frac{\Sigma_q}{\sigma^2}F(\overline{\mathbf{X}})^T + \Sigma_q\frac{\partial F(\overline{\mathbf{X}})}{\partial \sigma^2}\right)\Lambda\widehat{\boldsymbol{\alpha}} \tag{45}$$

- **Term 3**:

$$\frac{\partial F(\overline{\mathbf{X}})_{kl}}{\partial \sigma^2} = -\sigma \exp\left(\frac{(\mathbf{x}_k - \overline{\mathbf{x}}_l)^T(\mathbf{x}_k - \overline{\mathbf{x}}_l)}{-2\sigma^2}\right) \tag{46}$$

- **Term 4**:

$$\frac{\partial G(\overline{\mathbf{X}})_{kl}}{\partial \sigma^2} = -\sigma \exp\left(\frac{(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)^T(\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l)}{-2\sigma^2}\right) \tag{47}$$

Then, we use the updated terms 1-4 to compute the gradient $\frac{\partial L}{\partial \sigma^2}$ using the same procedure as described in section B.2.

C Augmentation Set Construction

Let $\mathbf{X}_u$ and $\mathbf{X}_t$ denote the sets of unlabeled candidate and training samples, respectively. The probability density of the two populations can be estimated as

$$p_u(\mathbf{x}) = \sum_{\mathbf{x}_n \in \mathbf{X}_u} k(\mathbf{x}, \mathbf{x}_n)/|\mathbf{X}_u| \tag{48}$$

$$p_t(\mathbf{x}) = \sum_{\mathbf{x}_n \in \mathbf{X}_t} k(\mathbf{x}, \mathbf{x}_n)/|\mathbf{X}_u| \tag{49}$$

Since we aim to identify unlabeled samples to inform the model areas in the space that are less explored by the training data, these samples should be from areas having a high density mass with respect to $p_u(\mathbf{x})$ but low density mass with respect to $p_t(\mathbf{x})$. The problem is formalized as:

$$\max_{\mathbf{A} \subseteq \mathbf{X}_u} \lambda \sum_{\mathbf{x} \in \mathbf{A}} p_u(\mathbf{x}) - \sum_{\mathbf{x} \in \mathbf{A}} p_t(\mathbf{x}) \tag{50}$$

The first term ensures that the selected area has abundant candidate data points to sample so that it has lower risk of containing isolated noise. The second term makes sure the selected region is not too close to the current training data. The optimal set is given by

$$\hat{\mathbf{A}} = \{\mathbf{x} | \lambda p_u(\mathbf{x}) - p_t(\mathbf{x}) > 0\} \tag{51}$$

where λ controls the size of $\hat{\mathbf{A}}$ for the given candidate and training datasets.

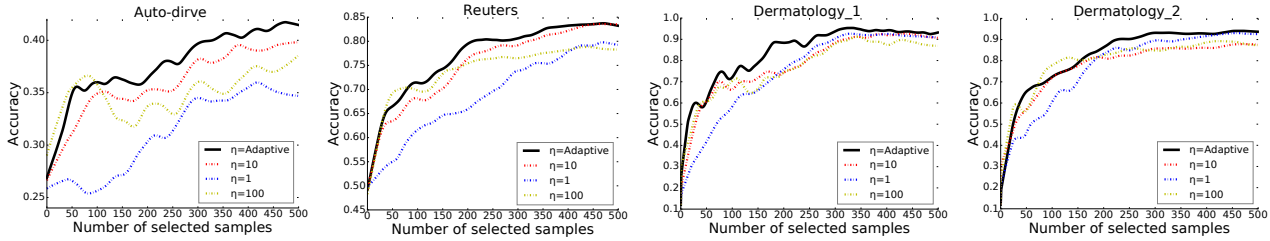
Posterior Augmentation. Once the augmented samples are identified, they will be used to augment the posterior distribution. Let $[K_{MA}]_{mi} = k(\overline{\mathbf{x}}_m, \hat{\mathbf{x}}_i)$, $K_{NA} = K_{NM}K_{MM}^{-1}K_{MA}$, and $[K_{AA}]_{ii'} = k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i'})$, the augmented covariance of MM-SGP defined by augmentation set $\hat{\mathbf{A}}$ is given by:

$$\Sigma_q^+ = \begin{bmatrix} \Sigma_q^{-1} & K_{MA} + K_{NM}^T K_{NA} \\ K_{MA}^T + K_{NA}^T K_{NM} & K_{AA} + K_{NA}^T K_{NA} \end{bmatrix}^{-1} \tag{52}$$

D Additional Real-Data Experiments

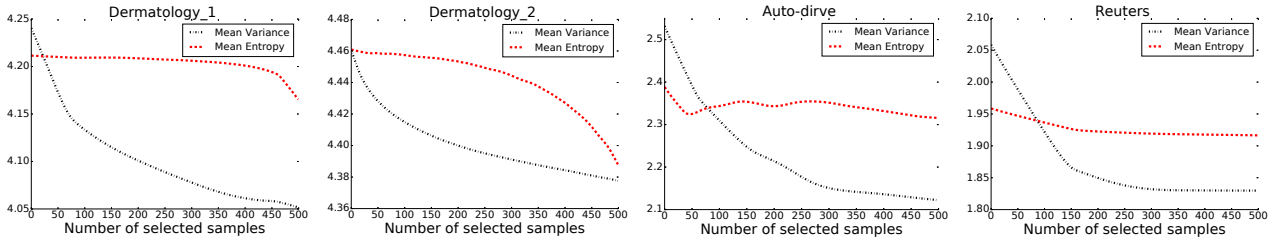
Impact of the balancing parameter: We provide the results on other datasets to demonstrate the impact the the balancing parameter η. A similar trend can be observed from Figures 5 and 6. First, entropy guided sampling can lead to slow convergence at the beginning of AL. It behaves more like random sampling when either the candidate pool or the cardinality of the label space is large (e.g. Auto-drive and Reuters). Second, the fixed balancing sampling ($\eta = 10$) has the performance close to optimal sampling (Adaptive). This is due to the fact that the mean predictive variance over the candidate pool is monotonically decreasing as AL proceeds. As a result, the variance will play less and less important role in active sampling as the model efficiently explores the data space. Third. the mean predictive entropy on the candidate pool does not decrease as significant as the variance especially in the beginning iterations of AL. There are two possible reasons for the slower decrease of the

entropy. One is that there may exist highly non-separable classes. In this case, adding samples near the decision boundaries (exploitation) may not help further reduce the predictive entropy in the corresponding region. In another case, the entropy is not deceasing since the model is exploiting the wrong decision boundaries resulted from the poor initialization of AL. For the entropy guided sampling, the model can not tell which case it is experiencing and the sampling can be less efficient if the second case happens frequently.



(a) (b) (c) (d)

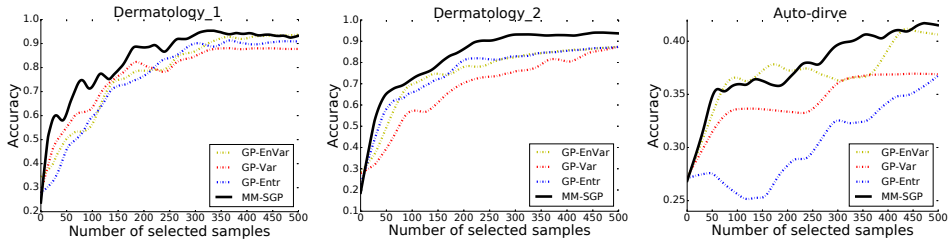
Figure 5: Impact of the balancing parameter η in data sampling



(a) (b) (c) (d)

Figure 6: Change of predictive variance and entropy

Impact of the MM constraints: We provide additional comparison results between the proposed MM-SGP with the standard GP on other datasets as shown in Figure 7. The results are consistent with the ones we have shown in Figure 4 (c) and (d), which further confirms the contribution of the maximum margin constraints. It is also worth to note that due to the large number of classes, the standard GP becomes extremely slow to complete the AL process on the Reuters dataset so the result is not included.

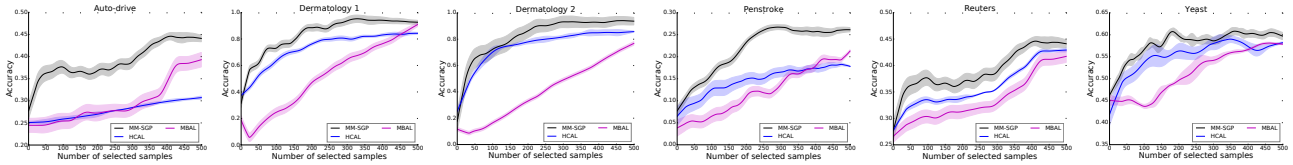


(a) (b) (c)

Figure 7: Impact of integrating MM constraints

Passive learning performance: We show the classification performance of the proposed MM-SGP in passive learning and compare with other kernel based models, including a standard GP, sparse GP, and SVM. Table 3 shows that MM-SGP has a robust prediction performance when the training size is small. This property makes MM-SGP a good candidate for active learning. Furthermore, MM-SGP achieves a similar prediction performance as GP and SVM when the training size becomes larger, indicating that the proposed active sampling method, rather than the classifier, contributes the most to the good active learning performance.

Compare with another two baselines: In Figure 8 we provide the result comparing MM-SGP with two specially designed AL models, Hierarchical clustering AL (HC-AL) [32] and margin based AL (MBAL). https://github.com/google/active-learning. The result shows that HC-AL achieves better AL performance than MBAL but clearly under-performs MM-SGP in all cases. MBAL appears to lack sufficient exploration especially in the early stage of AL.



(a) (b) (c) (d) (e) (f)

Figure 8: Compare with two additional baselines

Table 3: Passive Learning Performance Comparison

		Datasets					
Train Size	Method	Yeast	Penstroke	Auto-drive	Reuters	Dermatology I	Dermatology II
Initial	MM-SGP	0.46	0.07	0.3	0.55	0.08	0.11
	GP	0.47	0.07	0.3	0.5	0.08	0.1
	SGP	0.42	0.05	0.28	0.47	0.07	0.07
	SVM	0.42	0.06	0.27	0.51	0.08	0.1
100	MM-SGP	0.5	0.14	0.32	0.55	0.42	0.18
	GP	0.5	0.14	0.32	0.5	0.37	0.16
	SGP	0.48	0.12	0.31	0.48	0.3	0.15
	SVM	0.46	0.13	0.31	0.51	0.24	0.2
300	MM-SGP	0.51	0.22	0.35	0.57	0.59	0.5
	GP	0.52	0.2	0.33	0.52	0.58	0.48
	SGP	0.51	0.17	0.34	0.5	0.49	0.44
	SVM	0.49	0.19	0.35	0.53	0.55	0.46
500	MM-SGP	0.55	0.25	0.44	0.58	0.82	0.79
	GP	0.53	0.22	0.39	0.53	0.79	0.76
	SGP	0.52	0.2	0.41	0.5	0.77	0.69
	SVM	0.54	0.22	0.4	0.55	0.8	0.72
700	MM-SGP	0.58	0.28	0.47	0.63	0.93	0.9
	GP	0.57	0.26	0.45	0.57	0.92	0.89
	SGP	0.55	0.25	0.45	0.54	0.89	0.83
	SVM	0.56	0.24	0.46	0.64	0.93	0.91

Active sampling speed comparison: In Table 4, we compare the run-time of MM-SGP at 50% sparsity with three other methods with good AL performance. Our method maintains reasonable execution time among those top performing methods. It also shows desirable scalability as the size of the dataset increases.

Table 4: Active Sampling Time (seconds) Comparison

Trainsize	Dataset	Yeast	Penstroke	Reuters	Dermatology1	Dermatology2	Auto-drive
100	MM-SGP	1.3	6.6	37.1	3.7	2.8	2.7
	KMC	1.5	5.4	42.9	4.8	4.2	22
	VGP	0.9	1.7	25.4	3.4	2.6	2.2
	HC-AL	0.2	2.8	254	1.3	1.1	173
300	MM-SGP	6	25.7	110.5	23.6	17.7	12.1
	KMC	7.2	21.8	103.3	25.2	19.1	18.6
	VGP	4.7	13.2	78.6	20.9	14.3	10.7
	HC-AL	0.5	1.8	472	14.6	13.2	198
500	MM-SGP	41	83	263	202.1	98.1	48.6
	KMC	32.7	67.8	246	211	79.2	73.1
	VGP	20.5	40.2	183.5	162	143	111
	HC-AL	3.2	4.6	669	548	512	617

Source code: The code for MM-SGP and the datasets used in the experiments can be found: https://github.com/ritmininglab/Active-Learning-with-Maximum-Margin-Sparse-Gaussian-Processes.git