Spatially Smooth Bayesian FDR through Reproducing Kernel Hilbert Spaces

Anonymous Authors[1]

Abstract

Large-scale hypothesis testing is a cornerstone of science, yet it presents a fundamental challenge-the more hypotheses we test, the more false discoveries we find. To address this, methods controlling Family-Wise Error Rates (FWERs) have been developed, with the False Discovery Rate (FDR) emerging as the de facto standard. A significant challenge emerges when hypotheses have known relations: ignoring them invalidates standard FDR assumptions, while exploiting this structure substantially increases statistical power.

Here, we introduce a framework for spatial FDR control in continuous domains that, through graph kernel theory, unifies continuous and discrete settings for the first time. In contrast to current methods, we provide a convex optimization formulation that supports flexible regularization, enables principled hyperparameter selection via cross-validation, and is the first to infer prior null probabilities over the entire continuous space, not just at observed locations.

We validate our method on two setups: spatial locations derived from real-world anomaly detection datasets with generated p-values, and a differential gene expression task utilizing dependencies from protein-protein interaction graphs, where the FDR is evaluated against human-validated labels. In both cases, we demonstrate significant performance gains over state-of-the-art methods.

1. Introduction

The challenge of multiple hypothesis testing is one of the most fundamental challenges in science. It directly addresses our ability to control the probability of false discoveries that arise purely from random chance when simultaneously evaluating a large number of hypotheses. As our capacity to measure and collect high-dimensional data expands, effectively managing the statistical burden of multiple comparisons becomes a significant bottleneck, impeding our progress in translating abundant observations into meaningful scientific discoveries. In practice, this often translates to controlling some form of Family-Wise Error Rate(FWER), with the False Discovery Rate (FDR) being a widely adopted and powerful measure. For example, neuroimaging studies test millions of voxels across brain regions, genomic screens evaluate tens of thousands of genes for differential expression, and spatial transcriptomics maps thousands of molecular markers across tissue sections.

Importantly, in each of these domains hypotheses are not independent- nearby brain regions activate together, genes in the same pathway co-regulate, and spatially proximate cells share molecular signatures. In practice, while the relations structure of hypotheses is often overlooked, it is more the rule than the exception to encounter multiple hypothesis testing problems where individual hypotheses are not independent, significantly affecting lfdr results (). While classical FDR correction methods often focus on the case where FDR control holds for unknown dependencies(**?**) (see Section 2), there has been great interest in scenarios where these dependencies are indeed known or can be estimated (e.g., Tibshirani 2014, Xhai 2011). However, these methods typically require two conditions: (1) the dependencies must represent strict probabilistic relationships between hypotheses, and (2) these dependencies must be sparse. These requirements have led to a range of papers (**?**) demonstrating that, in general, this problem is only feasible under strict limitations on the number of local dependencies.

Another branch of research involves methods that address the dependency between hypotheses as a regularization term. The most common of these is SmoothFDR (2016), which addresses the problem where a general unweighted graph of association between hypotheses is given. In this approach, the lFDR control is regularized such that the prior probabilities (π) for the hypotheses are piecewise-constant over the given graph, achieved by regularizing the total variation (TV) over the graph (following Tibshirani 2014).

Despite their success, current graph-based smooth FDR methods face several fundamental limitations. First, they require discretizing continuous hypothesis spaces into graphs,

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

necessitating arbitrary choices about granularity and connectivity. Second, many graph topologies requires specialized algorithmic solutions, for example, methods designed for phylogenetic trees cannot be applied to gene ontology hierarchies, and approaches for spatial grids fail on network data. Third, Total Variation regularization enforces piecewise-constant solutions, which poorly model smoothly varying phenomena. While appropriate for segmentation tasks, this assumption is overly restrictive for many real-world applications where significance varies continuously across space. Fourth, these methods provide solutions only at observed graph nodes, with no principled framework for interpolating to unobserved locations. A critical limitation when sampling is sparse or when predictions are needed at new points.

In this paper, we address a much more general problem by assuming a continuous hypothesis space defined over some d-dimensional domain, then we show that almost any discrete problem with relations defined over a graph can be approximated under this framework. Importantly, by modeling the spatially-varying prior $\alpha(\text{loc})$ within a Reproducing Kernel Hilbert Space (RKHS), we move beyond the restrictive piecewise-constant assumptions of current L_1-based methods, enabling flexible smoothness regularization and the generation of continuous significance maps valid at both observed and unobserved locations. Our framework is computationally efficient, scaling as $O(N^2)$ via natural gradient optimization, and distinguishes itself by supporting rigorous likelihood-based hyperparameter selection with theoretical normalization guarantees. Ultimately, this work bridges spatial statistics and graph kernel theory, transforming discrete smooth FDR from a collection of topology-specific algorithms into a unified geometric problem: whether the structure is a spatial grid, a phylogenetic tree, or a protein interaction network, it is solved by a single framework where structure is encoded through the choice of kernel.

The remainder of this paper is organized as follows. Section 2 positions our work within the FDR literature. Section 3 establishes the problem formulation and spatially-varying mixture model, proving that component densities can be estimated independently of spatial structure (Proposition 3.1). Section 4 presents the point-wise optimization framework with natural gradient descent, demonstrating kernel cancellation that eliminates ill-conditioning (Lemma 4.1). Section 5 extends to entire-domain inference. Section 6 addresses hyperparameter selection via likelihood-based cross-validation (Proposition 6.1) and kernel selection for continuous domains and graph structures. Section 7 provides experimental validation on anomaly detection and gene expression datasets.

2. Related Work and Positioning

While the literature on FDR control is vast, it can be distinguished into two approaches: the global FDR and the local FDR (lFDR, or Bayesian FDR). In the lFDR framework, we assume that hypotheses are sampled from a mixture of two distributions: the null distribution, which is theoretically uniform (though often empirically estimated), and the alternative hypothesis distribution, which is typically unknown but assumed to be skewed toward zero. Controlling the FDR for related hypotheses is a long-standing challenge. Classical procedures, such as the Benjamini-Hochberg (B-H) method (**?**), control the FDR under independence or positive regression dependency, while more conservative approaches like the Benjamini-Yekutieli (B-Y) procedure (**?**) are required for arbitrary dependence. However, recent research has evolved from treating dependencies as a nuisance to leveraging them for increased statistical power. This has given rise to a rich field of structured FDR methods, which can be broadly categorized into several approaches.

Explicit Dependency Modeling. One major branch models dependencies as proper statistical relations (e.g., joint or conditional distributions, mutual information), mostly within the two-group model. These methods have shown that incorporating known dependency structures can lead to optimal ”oracle” procedures that significantly boost power (**??**). However, methods in this category (e.g., (**?**), (**?**)) typically require two restrictive conditions. First, the dependencies must represent actual probabilistic relationships where the dependency graph is given explicitly; this lacks mechanisms for handling continuous spatial domains or dense dependency structures without prior discretization. Second, these dependencies must be sparse; as noted by (**?**), general dependency modeling is often feasible only under strict limitations on the number of local dependencies.

Adaptive P-value Methods. A parallel line of work addresses spatial dependencies through adaptive p-value weighting. Methods such as LAWS (**?**) and STRAW (**?**) construct local weights from spatial neighbors, using discrete windows or local weighted averaging, to up-weight or down-weight p-values before applying standard FDR correction. Importantly, these approaches remain heuristic: the lack of explicit model of the underlying spatial structure provides no clear way to enforce prior beliefs or a principled mechanism for hyperparameter selection, and cannot interpolate to unobserved locations.

Regularization-Based Approaches. A second branch focuses on regularization-based approaches that treat dependency as a smoothness constraint. The *SmoothFDR* framework (**?**) regularizes the estimated prior null probabilities (π_0) to be smooth over a predefined graph by penalizing the Total Variation (TV) of the priors across graph edges. This graph-based paradigm faces fundamental limitations.

First, it requires the hypothesis space to be represented as a discrete graph, necessitating arbitrary discretization or clustering steps when the domain is continuous, explicitly making strong Markovian assumptions (known as the ”closed-world assumption,” where all dependencies of a single hypothesis are assumed to be encoded in the Markov blanket), rendering them impractical for many real-world scenarios. Moreover, they struggle with scenarios where the distribution of hypotheses contains many unobserved points or suffers from limited observability (e.g., geographical maps where only sparse signals of an underlying denser distribution are available).

Positioning and Contributions. While definitions may overlap, our framework is positioned under the FDR smoothing framework. This means that: (1) the spatial dependency is encoded trough regularization and the choice of kernel; and (2) we directly model the prior probability $\alpha(\text{loc})$ within the classical lFDR two-group mixture model, rather than reweighting p-values based on local criterion.

3. Problem Setup and Preliminaries

3.1. Preliminary Definitions

p-value is the probability under the null hypothesis of obtaining a test statistic at least as extreme as the one obtained. To illustrate with a coffee shop example, consider the sales under a new layout as an observed test statistic t. The p-value p would then be the probability of observing sales at least as high as t under the null distribution (i.e., the sales distribution of the old layout), assuming the new layout has no true effect (H_0). That is: $p = \Pr(T \geq t \mid H\_0)$.
Family-Wise Error Rate (FWER) is the probability of making one or more Type I errors (incorrectly rejecting a true null hypothesis at a given significance level α) when testing multiple hypotheses simultaneously. Following the previous example, if the coffee shop were to test 100 different layouts, for a significant level $p = 0.05$ we would expect 5 ”significant” layouts purely by luck.
False Discovery Rate (FDR) is the expected proportion of rejected hypotheses that are incorrectly rejected (i.e., false positives) in the multiple hypotheses scenario:

$$FDR = E\left[\frac{V}{R} \mid R \geq 0\right] P(R > 0)$$

where V is the number of false positives and R is the total number of rejections.

The **local false discovery rate (local-fdr)** represents an Bayesian approach to the multiple testing problem, offering distinct advantages over the global FDR (Efron, 2004). Instead of controlling an overall error rate, the local-fdr estimates the posterior probability that a *specific* hypothesis is null, given its observed test statistic or p-value. This provides a more granular and interpretable, case-by-case measure of significance. The core idea is to model the observed p-values as coming from a two-group mixture, with the local-fdr given test statistic z, is defined as :

$$\text{lfdr}(z) = P(H_0|Z = z) = \frac{\alpha_0 f_0(z)}{\alpha_0 f_0(z) + \alpha_1 f_1(z)}$$

Here, α_0 and α_1 are the prior probabilities of the null and alternative hypotheses, respectively ($\alpha_0 + \alpha_1 = 1$), and $f_0(z)$ and $f_1(z)$ are the probability density functions of the null and alternative distributions, respectively.

3.2. Problem Formulation and the Spatially-Varying Mixture Model.

We address the problem of multiple hypothesis testing where the hypotheses are indexed by a continuous space. Our goal is to estimate a spatially varying measure of significance, in this case the local False Discovery Rate (lFDR). Let the Hypothesis Index Space(HIS) be a continuous domain $\mathcal{H} \subset \mathbb{R}^D$. The statistical model $\mathcal{P}$ represents the set of all possible data-generating probability distributions. For each location loc $\in \mathcal{H}$, we define a null hypothesis $H_{\text{loc}} \subset \mathcal{P}$. For any true underlying distribution $P \in \mathcal{P}$, the set of true nulls is the (assumed measurable) subset of locations where the null hypothesis holds:

$$\mathcal{H}_0(P) := \{\text{loc} \in \mathcal{H} \mid P \in H_{\text{loc}}\}. \qquad (1)$$

We assume our observed data, consisting of p-values $\{p_i\}_{i=1}^N$ at discrete spatial coordinates $\{\text{loc}_i\}_{i=1}^N \subset \mathcal{H}$, are a finite sample from an underlying continuous p-value process, denoted $(p_{\text{loc}}(X))_{\text{loc}\in\mathcal{H}}$. Following (**?**), this **conceptual** process must satisfy the *Joint Measurability* condition which states that the process mapping $(\omega, \text{loc}) \mapsto p_{\text{loc}}(X(\omega))$ must be jointly measurable. Our method is designed for the general case of arbitrary relations, notably, **We do not require stronger assumptions such as Positive Regression Dependence on a Subset (PRDS).** Instead, the relation structure is modeled implicitly and flexibly through the choice of a reproducing kernel and smoothness regularization.

The natural way to model the p-value process according to the lFDR two-group mixture is by the marginal PDF:

$$f(p|\text{loc}) = \alpha(\text{loc}) f_0(p) + (1 - \alpha(\text{loc})) f_1(p) \qquad (2)$$

Where $\boldsymbol{\alpha}(\text{loc})$ is the *spatially-varying prior probability* that the null hypothesis H_{loc} is true. This is the core function we aim to estimate. Given alpha function as a mixing probability, a critical constrain is that $\alpha(\text{loc}) \in [0, 1]$ for all loc $\in \mathcal{H}$. $\boldsymbol{f_0(p)}$ is the PDF for p-values drawn from a location where the null hypothesis is true. While the validity assumption requires $f_0(p)$ to be the uniform distribution, we will not address it as such following (**?**), allowing for more flexible null distributions observed in practice. $\boldsymbol{f_1(p)}$ is the PDF for p-values drawn from a location where the **alternative hypothesis** is true.

Notice that when $\lambda_{\text{reg}} \to \infty$ (infinite smoothing), the optimal $\alpha(\text{loc}; \theta)$ becomes constant: $\alpha(\text{loc}; \theta) \equiv \bar{\alpha}$ for all loc. In this limit, our model reduces to the classical non-spatial two-group mixture, and the cross-validation criterion selects the marginal null probability $\bar{\alpha}$ that maximizes the marginal likelihood, exactly as in standard local FDR methods (**?**).

3.3. Estimating the Component Densities f_0 and f_1

A key advantage of our formulation is the separation between spatial structure and component densities. We assume the spatial dependence affects only the mixing proportion $\alpha(\text{loc})$, while the null and alternative densities, $f_0(z)$ and $f_1(z)$, remain spatially invariant. This enables a crucial decoupling of the estimation problem:

Proposition 3.1 (Marginal Density Independence)**.** *Under the spatially-varying mixture model (Eq. 2), the marginal density of test statistics follows the standard two-group mixture:*

$$f(z) = \bar{\alpha} f_0(z) + (1 - \bar{\alpha}) f_1(z) \tag{3}$$

where $\bar{\alpha} = \mathbb{E}_{loc}[\alpha(loc)]$ is the spatial average of the null probability.

Proof. See Supplementary Section B. □

This result confirms that the marginal distribution of test statistics, pooled across all locations, follows the classical local FDR model. Consequently, we can estimate f_0 and f_1 using established methods for non-spatial mixture models, such as the central matching or empirical null fitting methods of (**?**), treating spatial coordinates as irrelevant to the marginal estimation. The specific estimation procedure employed is detailed in Supplementary Section G.

3.4. The Optimization Problem

We frame the estimation of the spatially varying prior α as a Tikhonov-regularized maximum likelihood problem within a Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}_K$. Let $\|\cdot\|_{\mathcal{H}_K}$ denote the norm induced by the positive definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We seek the function $\alpha \in \mathcal{H}_K$ that minimizes the penalized negative log-likelihood:

$$\begin{aligned} \min_{\alpha \in \mathcal{H}_K} \quad \mathcal{J}(\alpha) = -\sum_{i=1}^{N} & \log\left(\alpha(\text{loc}_i) f_0(p_i)\right. \\ & \left. + (1 - \alpha(\text{loc}_i)) f_1(p_i)\right) \\ & + \lambda_{\text{reg}} \|\alpha\|_{\mathcal{H}_K}^2 \end{aligned} \tag{4}$$

subject to the pointwise constraints $0 \leq \alpha(\text{loc}_i) \leq 1$ for all $i \in \{1, \ldots, N\}$, with $\lambda_{\text{reg}} > 0$ controlling the regularization strength. Since the empirical risk term depends on α solely through its evaluations at the finite set of points $\{\text{loc}_i\}_{i=1}^N$, the Generalized Representer Theorem (**?**) guarantees that the minimizer lies in the finite-dimensional subspace spanned by the kernel sections centered at the data:

$$\alpha(\cdot) = \sum_{i=1}^{N} c_i K(\cdot, \text{loc}_i). \tag{5}$$

This result reduces the variational problem in (4) to a convex optimization over the coefficient vector $\mathbf{c} \in \mathbb{R}^N$. To solve this efficiently, we employ a **natural gradient** based optimization (**?**), which preconditions the update steps with the inverse kernel matrix to correct for the geometry of the hypothesis space (see Section 4). Also, while the framework admits any positive definite kernel, we focus on Matérn kernels, which allows Sobolev spaces of controlled smoothness. We discuss the implications of kernel choice and the smoothness parameter ν in Section **??**.

4. Point-wise Solution

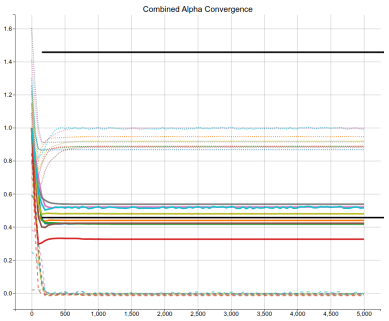
We begin by addressing the optimization problem at the observed locations $\{\text{loc}_i\}_{i=1}^N$ where we have observed test statistics $\{z_i\}_{i=1}^N$ (or equivalently, p-values $\{p_i\}_{i=1}^N$). Our goal then is to estimate $\alpha(\text{loc}_i)$ for each data point. The core challenge in solving Equation 4 is ensuring that $\alpha(\text{loc}_i) \in [0, 1]$ at all observed locations while maintaining a tractable optimization problem. Rather than imposing hard box constraints on $\alpha \in [0, 1]$, which would require constrained optimization methods, we introduce a *soft boundary penalty* that penalizes violations of the unit interval:

$$\Lambda_{\text{bound}}(\alpha) = \sum_{i=1}^{N} \left[\max(0, \alpha_i - 1)^2 + \max(0, -\alpha_i)^2\right] \tag{6}$$

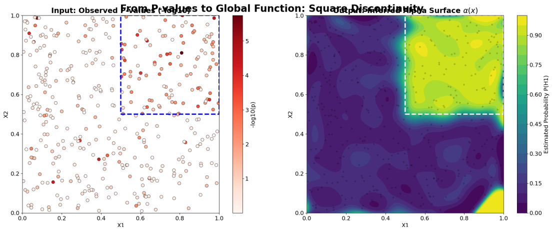
This formulation maintains several desirable properties. First, it is convex given each term is a composition of the convex squared hinge loss with the function values $\alpha(\text{loc}_i)$. Second, it is differentiable almost everywhere, enabling efficient gradient-based optimization. Third, the penalty is inactive when constraints are satisfied (i.e., when all $\alpha_i \in [0, 1]$), ensuring it does not interfere with well-behaved solutions. Then, our complete point-wise objective becomes:

$$\min_{\alpha \in \mathcal{H}_K} \mathcal{L}(\alpha) = -\sum_{i=1}^{N} \log h_i(\alpha) + \lambda_{\text{reg}} \|\alpha\|^2 + \lambda_{\text{bound}} \Lambda_{\text{bound}}(\alpha) \tag{7}$$

Alternative approaches include hard projection methods or squashing functions (e.g., logistic transformations); however, we found the penalty-based formulation to be both elegant and sufficient in practice. Figure **??** presents the minimum/maximum bounds (dashed lines) and mean (solid line) of α across optimization iterations for the all datasets evaluated in Section 7. As presented, α remains well-bounded within $[0, 1]$ from the beginning of the optimization.



(a) Alpha convergence across all datasets.



(b) Left: observed p-values. Right: α over the entire space.

Optimization via Natural Gradient Descent Following the Representer Theorem, we optimize over the coefficient vector $\mathbf{c} \in \mathbb{R}^N$ using gradient-based methods. However, standard optimizers operating in Euclidean geometry struggle with RKHS objectives: for smooth kernels (e.g., Matérn with large ν), the Gram matrix K has rapidly decaying eigenvalues, creating ill-conditioned optimization landscapes with narrow valleys. Even adaptive methods like Adam fail to compensate for the dense, structured correlations induced by the kernel. A more principled approach is the **natural gradient** (**?**), which accounts for the geometry of the parameter space. For RKHS optimization, the natural gradient is defined with respect to the metric tensor K (**?**):

$$\tilde{\nabla}_\alpha \mathcal{L} = K^{-1} \nabla_{\mathbf{c}} \mathcal{L} \tag{8}$$

The key insight is that this preconditioning eliminates the ill-conditioning entirely:

Lemma 4.1 (Kernel Cancellation in Natural Gradient)**.** *For the point-wise objective in Equation 7, the natural gradient admits the simple form:*

$$\tilde{\nabla}_\alpha \mathcal{L} = \mathbf{w} + 2\lambda_{reg}\mathbf{c} + \lambda_{bound} \nabla_\alpha \Lambda_{bound} \tag{9}$$

where $\mathbf{w}$ *is the residual vector with entries:*

$$w_i = -\frac{f_0(p_i) - f_1(p_i)}{\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)} \tag{10}$$

and $[\nabla_\alpha \Lambda_{bound}]_i = \begin{cases} 2(\alpha_i - 1) & \textit{if } \alpha_i > 1 \\ 2\alpha_i & \textit{if } \alpha_i < 0 \\ 0 & \textit{otherwise} \end{cases}$

Proof. See Supplementary Section C. □

This result is remarkable: the update rule depends only on residuals $\mathbf{w}$ and current parameters $\mathbf{c}$, **completely eliminating the ill-conditioned matrix K from the gradient step**. The kernel matrix is required only for forward evaluation $\boldsymbol{\alpha} = K\mathbf{c}$, which is numerically stable, which leads to significantly faster convergence.

5. Solution Over the Entire Domain

While the point-wise approach yields reliable estimates at observed coordinates, extending inference to the entire continuous domain $\mathcal{H}$ requires ensuring that $\alpha(\text{loc}) \in [0, 1]$ holds globally. This introduces a significant challenge: enforcing the bounds $\forall \text{loc} \in \mathcal{H}$ generates an infinite number of constraints. Even with the finite-dimensional coefficient vector provided by the Representer Theorem, verifying these constraints continuously across space is computationally intractable.Formally, this is a *Semi-Infinite Programming (SIP)* problem (**??**) , in Supplementary Section **??**, we provide a detailed analysis of classical SIP solvers (such as exchange methods and barrier functions) and demonstrate why they are ill-suited for the non-convex mixture landscape inherent to our problem. Consequently, we eschew direct SIP formulations in favor of a robust two-stage approach.

Rather than directly solving the SIP, we propose a solution that separates the problem into two stages, each addressing a specific aspect of the challenge. **The first stage** is the previously described Point-wise Estimation in which we solve the convex point-wise problem (Section 4) to obtain estimates $\{\hat{\alpha}_i\}_{i=1}^N$ at the observed locations $\hat{\boldsymbol{\alpha}}$. For the **second stage** we then treat the point-wise estimates $\{\hat{\alpha}_i\}_{i=1}^N$ as target labels and learn a globally valid function, hence, our goal is to find a function $\alpha(\text{loc}) \in [0, 1]$ that closely approximates these estimates across the entire domain. This is fundamentally a problem of *learning a probability-valued function*. From an information-theoretic perspective, this can be seen as finding a function $\alpha(\text{loc})$ that best approximates a target distribution $\hat{\alpha}(\text{loc})$ over the spatial domain. A natural measure of discrepancy between two probability distributions is the Kullback-Leibler (KL) divergence. For Bernoulli distributions with parameters $\hat{\alpha}$ and α :

$$D_{\text{KL}}(\hat{\alpha} \| \alpha) = \hat{\alpha} \log \frac{\hat{\alpha}}{\alpha} + (1 - \hat{\alpha}) \log \frac{1 - \hat{\alpha}}{1 - \alpha} \tag{11}$$

If we conceptualize the true spatial function $\alpha^*(\text{loc})$ as existing over the entire domain $\mathcal{H}$, and the point-wise estimates $\{\hat{\alpha}_i\}$ as observations of this function at discrete locations, then we seek to minimize:

$$\min_{\alpha \in \mathcal{H}_K} \int_{\mathcal{H}} D_{\text{KL}}(\alpha^*(\text{loc}) \| \alpha(\text{loc})) p(\text{loc}) \, d\text{loc} + \lambda_{\text{global}} \|\alpha\|^2_{\mathcal{H}_K} \tag{12}$$

where $p(\text{loc})$ is a measure over the domain. Minimizing KL divergence is equivalent to minimizing the cross-entropy (since the entropy of $\hat{\alpha}$ is constant), which leads naturally to the logistic loss. Notice the logistic loss is also a proper scoring rule (**?**), that is, minimized in expectation when $p = \hat{p}$, making it well-suited for matching probability-valued functions.

To ensure $\alpha(\text{loc}) \in [0, 1]$ everywhere while minimizing this discrepancy, we use a squashing function. The natural choice is the logistic function $\sigma(z) = \frac{1}{1+e^{-z}}$, which maps the entire real line to $[0, 1]$. This leads us to parameterize $\alpha(\text{loc}) = \sigma(g(\text{loc}))$ where $g \in \mathcal{H}_K$ is unconstrained. The resulting optimization is then the *kernel logistic regression*:

$$\begin{aligned}\mathbf{c}^* = \arg\min_{\mathbf{c}\in\mathbb{R}^N} - \sum_{i=1}^{N} [\hat{\alpha}_i \log \sigma(g_i) \\ +(1-\hat{\alpha}_i)\log(1-\sigma(g_i))] \\ + \lambda_{\text{global}}\mathbf{c}^T K\mathbf{c}\end{aligned} \tag{13}$$

Depends on the application, one may choose different regularization parameters λ_{reg} (Stage 1) and λ_{global} (Stage 2) to separately control the smoothness of the point-wise fit and the global interpolation. In practice, we typically set $\lambda_{\text{global}} \leq \lambda_{\text{reg}}$ to allow the global function to closely follow the point-wise estimates while maintaining spatial coherence. Another key practical consideration is how to initialize Stage 2. The coefficients $\mathbf{c}_{\text{point}}$ from Stage 1 provide a natural warm start, but they correspond to a different parameterization (direct α vs. squashed g). We use the inverse logistic transformation $\mathbf{c}^{(0)} = K^{-1} \cdot \text{logit}(\hat{\boldsymbol{\alpha}})$ where $\text{logit}(\alpha) = \log(\alpha/(1-\alpha))$ and we clip $\hat{\alpha}_i$ to $[\epsilon, 1-\epsilon]$ for small $\epsilon > 0$ (e.g., $\epsilon = 0.01$) to avoid numerical issues.

Crucially, the logistic loss formulation yields a **convex optimization problem**, avoiding the local minima issues inherent in direct squashing approaches (see Supplementary Section **??**). By applying the natural gradient framework derived in Section **??**, we obtain a remarkably simple update rule :

$$\tilde{\nabla}_{\mathbf{c}}\mathcal{L}_{\text{logistic}} = (\boldsymbol{\sigma} - \hat{\boldsymbol{\alpha}}) + 2\lambda_{\text{global}}\mathbf{c} \tag{14}$$

Figure 1b illustrates the alpha inference on a synthetic 2-dimensional dataset. The left panel shows the observed locations with p-values encoded by color, while the right panel presents the inferred alpha function. Notice that while the p-values are relatively noisy, our method was able to correctly infer the regions of high alpha.

6. Practical Considerations

6.1. Hyperparameter Selection via Cross-Validation

A key advantage of our framework is the ability to perform principled hyperparameter selection using standard likelihood-based cross-validation. This applies to all model parameters: the regularization weight λ_{reg}, kernel type, and kernel-specific parameters (e.g., length-scale ℓ and smoothness ν for Matérn kernels). More formally, let θ denote the complete hyperparameter vector and $p(\text{loc})$ the spatial sampling distribution, assumed independent of θ. The joint density factorizes as:

$$f(\text{loc}, z; \theta) = p(\text{loc}) \cdot f(z|\text{loc}; \theta) \tag{15}$$

Since $p(\text{loc})$ does not depend on θ, the cross-validation objective simplifies to maximizing the conditional log-likelihood on held-out test data:

$$\theta^* = \arg\max_{\theta} \sum_{i\in\text{Test}} \log f(z_i|\text{loc}_i; \theta) \tag{16}$$

For this approach to be valid, the conditional density $f(z|\text{loc}; \theta)$ must be properly normalized for all θ. This is guaranteed by the following result:

Proposition 6.1 (Hyperparameter-Independent Normalization)**.** *The joint density $f(loc, z; \theta)$ integrates to 1 for all hyperparameter values θ:*

$$\int_{\mathcal{H}} \int_{z} f(loc, z; \theta)\, dz\, dloc = 1, \quad \forall\theta \tag{17}$$

Consequently, the conditional density $f(z|loc; \theta)$ is a valid probability density for all θ, enabling rigorous likelihood-based model selection.

Proof. See Supplementary Section D. □

This result is crucial, it ensures that comparing test-set likelihoods across different hyperparameter settings is statistically meaningful. Unlike heuristic tuning approaches that rely on indirect metrics, our method directly optimizes the probability of observed data under the model, providing principled model selection with well-defined statistical interpretation. In practice, we employ 5-fold cross-validation and select hyperparameters via grid search over the parameter space (see Section 7 for details).

Moreover, the normalization constraint prevents pathological solutions where $\alpha(\text{loc}; \theta)$ is tuned to artificially inflate the marginal density $f(z)$ at the expense of spatial coherence. The mixture weights $\alpha(\text{loc}; \theta)$ must satisfy: $\int_{\mathcal{H}} \alpha(\text{loc}; \theta)\, p(\text{loc})\, d\text{loc} = \bar{\alpha}$, where $\bar{\alpha}$ is determined by the marginal data, not by θ.

6.2. Kernel Selection and Graph Kernels

Kernel Choice and Smoothness. The reproducing kernel $K(\cdot, \cdot)$ serves a dual role: it defines smoothness of $\alpha(x)$ through the RKHS norm and determines interpolation behavior between observed locations. For our framework, the kernel must be strictly positive definite to ensure invertibility of the Gram matrix (required for natural gradient optimization) and uniqueness of the solution. Beyond this, the kernel's differentiability controls the smoothness properties of functions in the RKHS via Sobolev space theory. Specifically, kernels generating Sobolev spaces $W^{m,2}(\mathbb{R}^d)$ with $m > d/2$ guarantee continuous, bounded functions-essential for reliable interpolation in the entire-domain setting. In our evaluations, we employ the **Matérn kernel family**, which provides explicit control over smoothness through the parameter $\nu > d/2$, balancing flexibility with theoretical guarantees (see Supplementary Section **??** for in-depth review).

Graph Kernels and Unified FDR Control. A significant conceptual advance of our framework is its natural extension to discrete graph structures via graph kernels. While many multiple testing problems arise on discrete topologies (gene regulatory networks, phylogenetic trees, gene ontology hierarchies), current smooth FDR methods require topology-specific algorithms. Our RKHS framework provides a unified solution: the *Graph Laplacian kernel* $K = L^\dagger$ enables our optimization algorithm to handle arbitrary graphs without modification. Crucially, this kernel's L_2 smoothness penalty $\|\alpha\|^2_{\mathcal{H}} = \sum_{(i,j)\in E} W_{ij}(\alpha_i - \alpha_j)^2$ contrasts fundamentally with the L_1 Total Variation penalty used in current methods, promoting smooth rather than piecewise-constant solutions. For hierarchical structures (e.g., gene ontologies with 50,000 terms but sparse sampling), **hyperbolic embeddings** offer compelling advantages: Sarkar's theorem (**?**) guarantees that any tree embeds into 2D hyperbolic space with arbitrarily low distortion, whereas Euclidean embeddings require $\Omega(\log n)$ distortion even in high dimensions. This enables our framework to handle hierarchical dependencies effectively even with 1% sampling rates (Supplementary Section **??** provides comprehensive treatment).

7. Evaluations

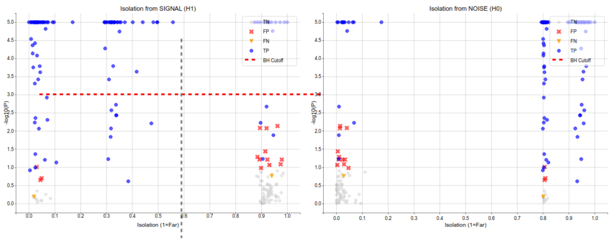


Figure 2. **Geometric Isolation Analysis.** Classification of hypotheses based on geometric distance from H_0 and H_1 clusters. The X-axis quantifies geometric isolation (0 = core, 1 = detached), while the Y-axis shows $-\log_{10}(p)$. Points within the H_1 core are rejected even at high p-values, demonstrating spatial coherence recovery.

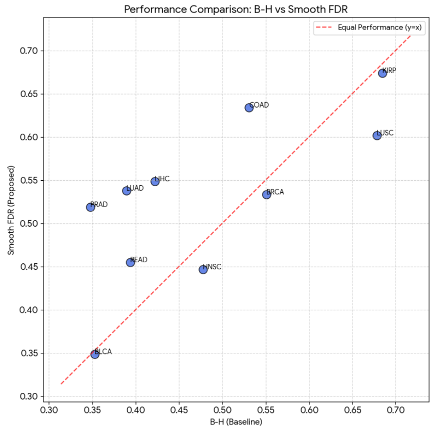
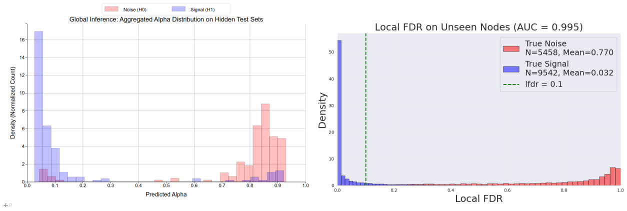


Figure 3. **TCGA Graph Kernel Performance.** Comparison of significant discoveries between Smooth-FDR and Benjamini-Hochberg (BH) across TCGA cancer cohorts. Our method consistently demonstrates superior statistical power across most datasets (PRAD, LIHC, LUSC), while maintaining robustness in cohorts where graph structure provides limited information (KIRP, BLCA).



(a) Spatial datasets

(b) TCGA datasets

Figure 4. **Global Inference on Unseen Locations.** Predicted $\alpha(\mathbf{x})$ values on held-out test points across (a) spatial anomaly detection datasets and (b) TCGA gene expression datasets. Clear separation between Signal (Blue) and Noise (Red) distributions demonstrates successful generalization to unobserved locations.

To evaluate the efficacy of the proposed method, we developed a semi-synthetic benchmarking protocol. While the spatial FDR literature has traditionally relied on 2D synthetic distributions (e.g., Gaussian blobs), these often diverge from the complex feature manifolds observed in real-world applications. Conversely, real datasets typically lack the ground truth required to define the null (H_0) or alternative (H_1) status of observations. To bridge this gap, we utilize 10 high-dimensional real-world datasets (e.g., *BreastW*, *Landsat*) to define realistic high-dimensional geometries, onto which we synthetically inject p-values.

To ensure the evaluation focuses on characterizeable geometric zones, we first perform manifold discovery by computing a Radial Basis Function (RBF) kernel matrix, identifying stable clusters, then to select $k = 3$ clusters that are mutually distant in kernel space. We assign specific roles to these clusters to define the underlying hypothesis structure: one serves as the primary source of non-null signals (C_0), while the remaining two represent null regions (C_1) and pure background noise (C_2). Finally, we introduce a Cluster Corruption parameter ($\gamma = 0.2$) that randomly flips the labels, creating "outlier" nulls and signals within otherwise homogeneous regions. P-values are generated based on these labels, with null cases (H_0) drawn from $\mathcal{U}[0, 1]$ and alternative cases (H_1) from a $\text{Beta}(0.05, 5)$ distribution.

Semi-Synthetic Spatial Benchmarking Protocol. For learning the ℓFDR on the described datasets, we utilized the Matérn kernel (rather than the RBF kernel used for clustering). We performed a cross-validation (CV) grid search to optimize the kernel parameters and the smoothness regularization parameter (λ_{reg}), with the α-bounds penalty regularizer set to $\lambda_{bound} = 500$. Supplementary Figure **??** presents the log-likelihood per hyperparameter selection for all datasets. Figure 2 presents the p-values ($-\log_{10} p$) against spatial distance from both the Signal (H_1) and Noise (H_0) cores over all datasets, with the X-axis quantifying **geometric isolation**, derived from the inverted normalized average kernel similarity to the k nearest neighbors within the reference cluster; values near 0 indicate locations deep within the core, while values near 1 imply spatial detachment. As the figure illustrates, points located within the geometric H_1 core are often rejected (labeled as H_1) even when exhibiting high p-values ($p \approx 0.67$), demonstrating the method's ability to recover spatially coherent signals. Conversely, observations within the H_0 core are effectively suppressed despite potential statistical fluctuations.

Generalization to Graph Structures: Gene Expression Analysis To validate generalization to arbitrary graph structures, we applied Smooth-FDR to differential gene expression analysis on The Cancer Genome Atlas (TCGA), with STRING (cite) as the source for relation between genes. This setting provides a natural testbed for spatial inference, as gene-gene assosication networks exhibit strong local dependencies that are often ignored by univariate correction methods. As a note, the HP tuning served as a great advantage, ensuring that the optimization learned a meaningful non-trivial diffusion map with out the need for any manual tunings. As Figure 3 shows, the method consistently demonstrates superior statistical power across the majority of datasets. In cohorts such as PRAD, LIHC, and LUSC, the Smooth-FDR method yields a markedly higher detection rate compared to the baseline. Furthermore, in datasets where the graph structure offered limited informative value (e.g., KIRP, BLCA), the method exhibited robustness, converging to a solution comparable to the baseline without inflating the false discovery rate.

Evaluation of Global Probability Inference. To validate the generalization of the learned null probability function $\alpha(\mathbf{x})$ to unobserved locations, we performed a stratified 80/20 hold-out evaluation across all datasets. We aggregated the predicted α values on the test sets to quantify the separation between known signal (H_1) and noise (H_0) points. Figure 4a displays the resulting histograms, demonstrating a clear and robust separation between the classes. Finally, we applied our framework to the TCGA gene expression datasets. We randomly masked the p-values of 20% of genes and inferred their latent signal probability α. Figure 4b shows the aggregated predictions across all TCGA cohorts, presenting overwhelming successes in recovering the true signal/noise separation for held-out genes. This demonstrates our framework's ability to perform meaningful spatial inference on complex biological networks with real-world data.

8. Conclusions

We introduced a unified framework for spatial FDR control that transforms the problem from requiring topology-specific algorithms into a geometric optimization where spatial structure is encoded through kernel choice. Our approach provides the first method to produce continuous significance maps valid across entire domains with convex optimization guarantees, advancing beyond current methods through flexible smoothness regularization, principled hyperparameter selection, and $O(N^2)$ computational scaling. Experimental validation demonstrates substantial power gains while maintaining robust FDR control and successful generalization to unobserved locations.

By unifying smooth FDR methods with graph kernel theory, our framework opens promising directions. Specifically, hyperbolic embeddings, recently emerging as a powerful tool for graph kernels, offer compelling advantages for diverse fields, allowing for the use of sophisticated structures like hierarchical graphs, temporal graphs, and more.

References

A. Proof of Marginal Density Independence

Here we provide the full derivation supporting Proposition 3.1.

Proof of Proposition 3.1. Consider the observed locations $\{\text{loc}_i\}_{i=1}^N$ as samples drawn from a spatial sampling distribution with density $p(\text{loc})$ over the domain $\mathcal{H}$. The marginal density of a test statistic z, denoted $f(z)$, is obtained by integrating the conditional mixture model (Eq. 2 in main text) over the spatial domain:

$$\begin{aligned} f(z) &= \int_{\mathcal{H}} f(z \mid \text{loc})\, p(\text{loc})\, d\text{loc} && (18) \\ &= \int_{\mathcal{H}} [\alpha(\text{loc}) f_0(z) + (1-\alpha(\text{loc})) f_1(z)]\, p(\text{loc})\, d\text{loc} && (19) \end{aligned}$$

Since $f_0(z)$ and $f_1(z)$ do not depend on loc (by our spatially-invariant assumption), we can factor them out of the integral:

$$\begin{aligned} f(z) = f_0(z) &\int_{\mathcal{H}} \alpha(\text{loc})\, p(\text{loc})\, d\text{loc} \\ &+ f_1(z) \int_{\mathcal{H}} (1-\alpha(\text{loc}))\, p(\text{loc})\, d\text{loc} && (20) \end{aligned}$$

Define the global average null probability as:

$$\bar{\alpha} = \mathbb{E}_{\text{loc}}[\alpha(\text{loc})] = \int_{\mathcal{H}} \alpha(\text{loc})\, p(\text{loc})\, d\text{loc} \tag{21}$$

Then the second integral becomes:

$$\begin{aligned} \int_{\mathcal{H}} (1-\alpha(\text{loc}))\, p(\text{loc})\, d\text{loc} &= \int_{\mathcal{H}} p(\text{loc})\, d\text{loc} - \int_{\mathcal{H}} \alpha(\text{loc})\, p(\text{loc})\, d\text{loc} \\ &= 1 - \bar{\alpha} && (22) \end{aligned}$$

Substituting Equations (21) and (22) into Equation (20):

$$f(z) = \bar{\alpha} f_0(z) + (1-\bar{\alpha}) f_1(z) \tag{23}$$

This is precisely the standard two-group mixture form used in classical local FDR methods, confirming that the spatial structure affects only the mixing proportion $\bar{\alpha}$, not the functional forms of f_0 and f_1. □

B. Derivation of Natural Gradient for Point-wise Optimization

Here we provide the complete derivation supporting Lemma 4.1 from the main text.

B.1. Standard Euclidean Gradient

The objective function (Eq. 7 in main text) decomposes into three terms:

$$\mathcal{L}(\mathbf{c}) = \mathcal{L}_{\text{data}}(\mathbf{c}) + \mathcal{L}_{\text{reg}}(\mathbf{c}) + \mathcal{L}_{\text{bound}}(\mathbf{c}) \tag{24}$$

We compute the gradient of each term separately.

B.1.1. Data Term Gradient

The negative log-likelihood term is:

$$\mathcal{L}_{\text{data}} = -\sum_{i=1}^{N} \log\left[\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)\right] \tag{25}$$

Using the chain rule with $\alpha_i = (K\mathbf{c})_i = \sum_{j=1}^{N} K_{ij} c_j$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{data}}}{\partial c_j} &= -\sum_{i=1}^{N} \frac{1}{\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)} \cdot \frac{\partial}{\partial c_j}\left[\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)\right] && (26)\\
&= -\sum_{i=1}^{N} \frac{f_0(p_i) - f_1(p_i)}{\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)} \cdot \frac{\partial \alpha_i}{\partial c_j} && (27)\\
&= -\sum_{i=1}^{N} \frac{f_0(p_i) - f_1(p_i)}{\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)} \cdot K_{ij} && (28)
\end{aligned}$$

In vector notation, define the residual vector $\mathbf{w} \in \mathbb{R}^N$ with entries:

$$w_i = -\frac{f_0(p_i) - f_1(p_i)}{\alpha_i f_0(p_i) + (1-\alpha_i) f_1(p_i)} \tag{29}$$

Then:

$$\nabla_{\mathbf{c}} \mathcal{L}_{\text{data}} = K\mathbf{w} \tag{30}$$

B.1.2. Regularization Term Gradient

The RKHS regularization is:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \|\alpha\|_{\mathcal{H}_K}^2 = \lambda_{\text{reg}} \mathbf{c}^T K \mathbf{c} \tag{31}$$

Taking the gradient:

$$\nabla_{\mathbf{c}} \mathcal{L}_{\text{reg}} = 2\lambda_{\text{reg}} K\mathbf{c} \tag{32}$$

B.1.3. Boundary Penalty Gradient

The boundary penalty is:

$$\Lambda_{\text{bound}}(\alpha) = \sum_{i=1}^{N} \left[\max(0, \alpha_i - 1)^2 + \max(0, -\alpha_i)^2\right] \tag{33}$$

Define the element-wise gradient:

$$[\nabla_\alpha \Lambda_{\text{bound}}]_i = \frac{\partial}{\partial \alpha_i}\left[\max(0, \alpha_i - 1)^2 + \max(0, -\alpha_i)^2\right] \tag{34}$$

This has the closed form:

$$[\nabla_\alpha \Lambda_{\text{bound}}]_i = \begin{cases} 2(\alpha_i - 1) & \text{if } \alpha_i > 1 \\ 2\alpha_i & \text{if } \alpha_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Using the chain rule $\frac{\partial \Lambda_{\text{bound}}}{\partial c_j} = \sum_{i=1}^{N} \frac{\partial \Lambda_{\text{bound}}}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial c_j}$:

$$\nabla_{\mathbf{c}} \mathcal{L}_{\text{bound}} = \lambda_{\text{bound}} K \nabla_\alpha \Lambda_{\text{bound}} \tag{36}$$

B.2. Combined Gradient and Factorization

Combining Equations (30), (32), and (36):

$$\begin{aligned}\nabla_{\mathbf{c}}\mathcal{L} &= K\mathbf{w} + 2\lambda_{\text{reg}}K\mathbf{c} + \lambda_{\text{bound}}K\nabla_{\alpha}\Lambda_{\text{bound}} &\quad (37)\\ &= K\left[\mathbf{w} + 2\lambda_{\text{reg}}\mathbf{c} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}}\right] &\quad (38)\end{aligned}$$

This factorization reveals that the Gram matrix K appears as a leading factor, which is the source of the ill-conditioning in standard gradient descent.

B.3. Natural Gradient and Kernel Cancellation

Proof of Lemma 4.1. The natural gradient is defined as:

$$\tilde{\nabla}_{\alpha}\mathcal{L} = K^{-1}\nabla_{\mathbf{c}}\mathcal{L} \quad (39)$$

Substituting the factored form from Equation (38):

$$\begin{aligned}\tilde{\nabla}_{\alpha}\mathcal{L} &= K^{-1}\left(K\left[\mathbf{w} + 2\lambda_{\text{reg}}\mathbf{c} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}}\right]\right) &\quad (40)\\ &= \left(K^{-1}K\right)\left[\mathbf{w} + 2\lambda_{\text{reg}}\mathbf{c} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}}\right] &\quad (41)\\ &= I\left[\mathbf{w} + 2\lambda_{\text{reg}}\mathbf{c} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}}\right] &\quad (42)\\ &= \mathbf{w} + 2\lambda_{\text{reg}}\mathbf{c} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}} &\quad (43)\end{aligned}$$

where we used $K^{-1}K = I$ (the identity matrix). This completes the proof. □

The complete natural gradient descent update at iteration k is:

Algorithm 1 Natural Gradient Update for Point-wise FDR

```
Input: Current coefficients $\mathbf{c}^{(k)}$, learning rate $\eta$
Compute $\boldsymbol{\alpha}^{(k)} = K\mathbf{c}^{(k)}$ $\{$Forward pass: $O(N^2)\}$
Compute residuals: $w_i^{(k)} = -\frac{f_0(p_i)-f_1(p_i)}{\alpha_i^{(k)}f_0(p_i)+(1-\alpha_i^{(k)})f_1(p_i)}$ $\{O(N)\}$
Compute boundary gradients: $[\nabla_{\alpha}\Lambda_{\text{bound}}]_i$ $\{O(N)\}$
Form natural gradient:
    $\tilde{\nabla}^{(k)} = \mathbf{w}^{(k)} + 2\lambda_{\text{reg}}\mathbf{c}^{(k)} + \lambda_{\text{bound}}\nabla_{\alpha}\Lambda_{\text{bound}}$ $\{O(N)\}$
Update: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k)} - \eta\tilde{\nabla}^{(k)}$ $\{O(N)\}$
Return: $\mathbf{c}^{(k+1)}$ =0
```

The total per-iteration complexity is $O(N^2)$, dominated by the kernel matrix-vector product in the forward pass.

C. Proof of Hyperparameter-Independent Normalization

Here we provide the complete proof of Proposition 6.1 from the main text, which validates the use of likelihood-based cross-validation for hyperparameter selection.

Proof of Proposition 6.1. Let $\mathcal{H}$ denote the spatial domain and $[z_{\min}, z_{\max}]$ denote the support of the test statistics. We must show that the normalization constant:

$$Z(\theta) = \int_{\mathcal{H}}\int_{z_{\min}}^{z_{\max}} f(\text{loc}, z; \theta)\, dz\, d\text{loc} \quad (44)$$

equals 1 for all hyperparameter values θ.

By the factorization $f(\text{loc}, z; \theta) = p(\text{loc}) \cdot f(z|\text{loc}; \theta)$ and the mixture model (Eq. 2 in main text):

$$\begin{aligned} Z(\theta) &= \int_{\mathcal{H}} \int_{z_{\min}}^{z_{\max}} p(\text{loc}) \cdot f(z|\text{loc}; \theta)\, dz\, d\text{loc} \\ &= \int_{\mathcal{H}} p(\text{loc}) \int_{z_{\min}}^{z_{\max}} [\alpha(\text{loc}; \theta) f_0(z) + (1 - \alpha(\text{loc}; \theta)) f_1(z)]\, dz\, d\text{loc} \end{aligned} \tag{45}$$

Since $f_0(z)$ and $f_1(z)$ do not depend on loc (by assumption), we can factor them out of the inner integral:

$$\begin{aligned} Z(\theta) = \int_{\mathcal{H}} p(\text{loc}) \Bigg[&\alpha(\text{loc}; \theta) \underbrace{\int_{z_{\min}}^{z_{\max}} f_0(z)\, dz}_{\text{Term A}} \\ &+ (1 - \alpha(\text{loc}; \theta)) \underbrace{\int_{z_{\min}}^{z_{\max}} f_1(z)\, dz}_{\text{Term B}} \Bigg] d\text{loc} \end{aligned} \tag{46}$$

Since f_0 and f_1 are valid probability density functions over $[z_{\min}, z_{\max}]$, by definition they integrate to 1:

$$\text{Term A:} \quad \int_{z_{\min}}^{z_{\max}} f_0(z)\, dz = 1 \tag{47}$$

$$\text{Term B:} \quad \int_{z_{\min}}^{z_{\max}} f_1(z)\, dz = 1 \tag{48}$$

Substituting into Equation (46):

$$\begin{aligned} Z(\theta) &= \int_{\mathcal{H}} p(\text{loc})\, [\alpha(\text{loc}; \theta) \cdot 1 + (1 - \alpha(\text{loc}; \theta)) \cdot 1]\, d\text{loc} \\ &= \int_{\mathcal{H}} p(\text{loc})\, [\alpha(\text{loc}; \theta) + 1 - \alpha(\text{loc}; \theta)]\, d\text{loc} \\ &= \int_{\mathcal{H}} p(\text{loc})\, d\text{loc} \end{aligned} \tag{49}$$

Since $p(\text{loc})$ is itself a probability density function over the spatial domain $\mathcal{H}$, by definition:

$$\int_{\mathcal{H}} p(\text{loc})\, d\text{loc} = 1 \tag{50}$$

Finally, combining Equations (49) and (50):

$$Z(\theta) = 1, \quad \forall \theta \tag{51}$$

Crucially, this derivation makes **no reference to the specific functional form of $\alpha(\mathbf{loc}; \theta)$** beyond the requirement that $\alpha : \mathcal{H} \times \Theta \to [0, 1]$. The normalization holds regardless of:

- The choice of kernel K
- The regularization strength λ_{reg}
- Kernel hyperparameters (ℓ, ν, etc.)
- The learned coefficient vector $\mathbf{c}$

This completes the proof. □

C.1. Implications for Cross-Validation

Proposition 6.1 has several important consequences for hyperparameter selection:

Valid likelihood comparisons. Since $f(\text{loc}, z; \theta)$ is normalized for all θ, the conditional density satisfies:

$$\int_z f(z|\text{loc}; \theta)\, dz = \frac{\int_z f(\text{loc}, z; \theta)\, dz}{p(\text{loc})} = \frac{p(\text{loc})}{p(\text{loc})} = 1 \tag{52}$$

This means test-set log-likelihoods $\sum_{i\in\text{Test}} \log f(z_i|\text{loc}_i; \theta)$ are directly comparable across different θ values without any normalization corrections.

The normalization constraint prevents pathological solutions where $\alpha(\text{loc}; \theta)$ is tuned to artificially inflate the marginal density $f(z)$ at the expense of spatial coherence. The mixture weights $\alpha(\text{loc}; \theta)$ must satisfy:

$$\int_{\mathcal{H}} \alpha(\text{loc}; \theta)\, p(\text{loc})\, d\text{loc} = \bar{\alpha} \tag{53}$$

where $\bar{\alpha}$ is determined by the marginal data, not by θ.

Consistency with classical FDR. When $\lambda_{\text{reg}} \to \infty$ (infinite smoothing), the optimal $\alpha(\text{loc}; \theta)$ becomes constant: $\alpha(\text{loc}; \theta) \equiv \bar{\alpha}$ for all loc. In this limit, our model reduces to the classical non-spatial two-group mixture, and the cross-validation criterion selects the marginal null probability $\bar{\alpha}$ that maximizes the marginal likelihood—exactly as in standard local FDR methods (**?**).

C.2. Practical Cross-Validation Procedure

In our experiments (Section 7 in main text), we employ the following procedure:

Algorithm 2 Hyperparameter Selection via Cross-Validation

```
Require: Data $\{(p_i, \text{loc}_i)\}_{i=1}^N$, hyperparameter grid $\Theta$, number of folds $K$
Ensure: Optimal hyperparameters $\theta^*$
Partition data into $K$ folds: $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K$
for each $\theta \in \Theta$ do
    Initialize cumulative log-likelihood: $\mathcal{L}_{\text{CV}}(\theta) = 0$
    for fold $k = 1, \ldots, K$ do
        Train on $\mathcal{D}^{(k)}_{\text{train}} = \mathcal{D} \setminus \mathcal{D}_k$
        Obtain coefficients $\mathbf{c}^{(k)}(\theta)$ by solving Eq. 7
        Compute test log-likelihood:
          $\mathcal{L}_k(\theta) = \sum_{i\in\mathcal{D}_k} \log f(z_i|\text{loc}_i; \theta, \mathbf{c}^{(k)})$
        Update: $\mathcal{L}_{\text{CV}}(\theta) \leftarrow \mathcal{L}_{\text{CV}}(\theta) + \mathcal{L}_k(\theta)$
    end for
end for
return $\theta^* = \arg\max_{\theta\in\Theta} \mathcal{L}_{\text{CV}}(\theta)$ =0
```

Computational cost. For a grid of size $|\Theta|$ and K folds, the total number of optimizations is $|\Theta| \times K$. Each optimization solves the point-wise problem (Eq. 7 in main text) with complexity $O(TN^2)$ where T is the number of gradient steps. In practice, with $|\Theta| \approx 20$, $K = 5$, $T \approx 50$, and $N \approx 500$, the entire cross-validation procedure completes in under 10 minutes on a standard laptop.

Grid design. We typically search over:

- Regularization: $\lambda_{\text{reg}} \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$
- Kernel length-scale: $\ell \in \{0.1, 0.5, 1.0, 2.0, 5.0\} \times \text{median}(\|\text{loc}_i - \text{loc}_j\|)$

- Smoothness (Matérn): $\nu \in \{1.5, 2.5, 3.5\}$ (ensuring $\nu > d/2$ for continuity)

The boundary penalty coefficient λ_{bound} is typically fixed at a large value (e.g., 500) to ensure strict constraint satisfaction.

D. Marginal Density Estimation

Motivation. The estimation of the marginal component densities, f_0 and f_1, typically follows one of two paradigms, each with distinct limitations. The first approach assumes the inputs are well-calibrated p-values, enforcing a strict Uniform distribution for the null hypothesis ($f_0 \sim U[0, 1]$). While theoretically sound, this direction fails fundamentally when the input data are z-scores or other test statistics, and the calibration itself is rare in practice (**?**).

The second approach operates on z-values (or generic continuous data) and relies on general mixture models to separate the null and alternative distributions based on their shape. One might assume this direction could handle p-values by simply treating them as bounded data points. However, a nuanced problem arises: when the null hypothesis is truly Uniform, it manifests as a highly ”overdispersed” or maximal-entropy background relative to the signal. Standard clustering or separation algorithms, which typically expect compact modes for both classes, fail to identify the signal against this flat, featureless background. This dilemma, where p-value methods cannot handle z-scores, and z-score methods fail on p-values due to null overdispersion, motivated many hybrid strategies, which we did not dive into here.

For the evaluations in this paper, we employ a simple hybrid strategy which respect the theoretical properties of the p-value domain to define the null, while utilizing the z-score domain to characterize the alternative signal. First, we strictly enforce the theoretical null hypothesis, setting $f_0(p) = 1$ for all $p \in [0, 1]$. This avoids the estimation instability caused by the overdispersed null. Second, to estimate the alternative density f_1, we transform the p-values into probit space (z-scores) via the inverse standard normal CDF, $z_i = \Phi^{-1}(1 - p_i)$. In this space, the alternative distribution is well-approximated by a Gaussian. We isolate the ”signal” tail by selecting observations with $p_i < 0.2$ and estimate the alternative parameters (μ_1, σ_1) using the sample moments of these tail z-scores:

$$\hat{\mu}_1 = \text{mean}(z \mid p < 0.2), \quad \hat{\sigma}_1 = \text{std}(z \mid p < 0.2) \tag{54}$$

The alternative density is then defined in z-space as $\mathcal{N}(\hat{\mu}_1, \hat{\sigma}_1)$ and transformed back to p-value space using the appropriate Jacobian:

$$f_1(p) = \frac{\phi(z_p \mid \hat{\mu}_1, \hat{\sigma}_1)}{\phi(z_p \mid 0, 1)}, \quad \text{where } z_p = \Phi^{-1}(1 - p) \tag{55}$$

This procedure yields a robust marginal model that combines the stability of the theoretical null with the flexibility of a parametric alternative fit.

E. Kernel Selection Principles

Having established both point-wise (Section **??**) and entire-domain (Section **??**) solution approaches, we now address the question of kernel choice. The reproducing kernel $K(\cdot, \cdot)$ serves a dual role: it defines the smoothness of $\alpha(x)$ through the RKHS norm $\|\alpha\|^2_{\mathcal{H}_K} = c^T K c$, and it determines the interpolation behavior between observed locations-essential for predictions at new points in the entire-domain setting. A fundamental requirement for any symmetric function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to serve as a reproducing kernel is positive definiteness. For our specific objective function with a logarithmic loss term $\log(\alpha(x_i))$, a slightly stronger condition is necessary. Since we employ natural gradient optimization, which requires computing $K^{-1}\nabla_{\mathbf{c}}\mathcal{L}$ (see Section **??**), the Gram matrix K must be strictly positive definite ($K \succ 0$) and therefore invertible. This also ensures uniqueness of the solution in coefficient space, which is important given the logarithm's barrier-like behavior near zero. Most standard kernels (Gaussian, Matérn) satisfy this property when evaluated on distinct points.

While positive definiteness guarantees a valid RKHS, the *differentiability* of the kernel determines the smoothness properties of functions in that space. This relationship is formalized through Sobolev space theory. A Sobolev space $W^{m,2}(\mathbb{R}^d)$ contains functions whose derivatives up to order m are square-integrable, where m quantifies smoothness. An RKHS can often be identified with a specific Sobolev space, with the kernel's differentiability determining m. A key result is the Sobolev embedding theorem is that if $m > d/2$, then functions in $W^{m,2}(\mathbb{R}^d)$ are guaranteed to be continuous and bounded.

The smoothness requirements depend on the application. If the goal is solely to estimate $\alpha(\text{loc}_i)$ at observed locations, kernel differentiability is not strictly required- the optimization machinery and Representer Theorem only require kernel evaluation at data points. However, when interpolating to new locations in the entire-domain setting, kernel differentiability becomes essential. By choosing a smooth kernel satisfying $m > d/2$, we impose the prior belief that $\alpha(x)$ varies continuously across space, ensuring the function behaves predictably between observed points. For example, the linear kernel $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$ is a valid choice for point-wise estimation, where it effectively uses the correlation between location vectors as a similarity measure. However, for entire-domain generalization, the linear kernel is constrained by its limited smoothness properties. Given our need for a flexible, configurable kernel suitable for both point-wise and entire-domain settings, in our evaluations we employ the **Matérn kernel family** which includes a smoothness parameter $\nu > 0$ that explicitly controls the differentiability. To satisfy the Sobolev embedding condition for d-dimensional spaces, we require $\nu > d/2$.

E.1. Graph Kernels: Bridging Discrete and Continuous FDR

A significant conceptual advance of our framework is the connection it establishes between smooth FDR methods and graph kernel theory. While we focused on continuous spatial domains $\mathcal{H} \subset \mathbb{R}^d$, many multiple testing problems arise on discrete structures: gene regulatory networks, phylogenetic trees, gene ontology hierarchies, and social networks. Indeed, most current smooth FDR methods operate on such discrete graph representations, requiring topology-specific algorithmic solutions where methods designed for trees cannot be applied to scale-free networks, and techniques for hierarchical structures fail on general graphs. Our RKHS framework provides a unified solution: by choosing an appropriate graph kernel, the same optimization algorithm (Section **??**) applies to arbitrary discrete topologies.

For a comprehensive treatment of graph kernels, we refer the reader to the seminal work of Kondor and Lafferty (2002) and its subsequent extensions by Smola and Kondor (2003). Within this framework, the *Graph Laplacian kernel* ($K = L^\dagger$) serves as a canonical example. Its associated RKHS norm, $\|\alpha\|^2_{\mathcal{H}} = \boldsymbol{\alpha}^T L \boldsymbol{\alpha} = \sum_{(i,j)\in E} W_{ij}(\alpha_i - \alpha_j)^2$, explicitly penalizes signal differences across edges using an L_2 metric. This formulation contrasts fundamentally with the Total Variation (TV) penalty ($\sum W_{ij}|\alpha_i - \alpha_j|$) employed by current graph-based FDR methods, whereas TV relies on the L_1 norm to enforce piecewise-constant clustering, the Laplacian kernel promotes smooth variation across the network. To capture longer-range dependencies, this approach naturally extends to *diffusion kernels* ($K = \exp(-\beta L)$) and *random walk kernels*, which incorporate global graph topology and allow distant but well-connected nodes to influence local estimation. A particularly promising direction involves the use of **Hyperbolic Embeddings for Hierarchical Structures**, as many biological and social networks exhibit inherent hierarchical organization : gene ontologies (GO), phylogenetic trees, and protein interaction networks with hub-spoke patterns. For such structures, the seminal result of (**?**) regarding hyperbolic spaces applies. Briefly, Sarkar's Theorem states that any tree with n nodes can be embedded into the 2-dimensional hyperbolic space (Poincaré disk $\mathbb{H}^2$) with arbitrarily low distortion $(1 + \epsilon)$ for any $\epsilon > 0$.This dimension efficiency is crucial for practical application, but more fundamentally, it addresses a geometric incompatibility. Consider gene ontology enrichment testing containing 50,000 terms (**?**), but observations at only 500 locations (1% sampling). In Euclidean space, such hierarchical structures face an intrinsic "capacity" problem: trees exhibit exponential volume growth, whereas Euclidean space has only polynomial growth. Consequently, embedding a tree into *any* low-dimensional Euclidean space $\mathbb{R}^d$ must incur significant distortion. Even high-dimensional Euclidean embeddings fail to capture the tree topology faithfully compared to the hyperbolic plane. **In contrast**, by leveraging Sarkar's construction, we can embed these dependencies into just 2 dimensions with near-perfect fidelity, effectively bypassing the limitations of Euclidean kernels for hierarchical data.

F. Alternative Approaches for Enforcing $\alpha(x) \in [0, 1]$ Over Continuous Domains

Before developing the convex two-step kernel logistic regression framework, we explored several direct methods for enforcing $\alpha(\text{loc}) \in$ [7] across the entire continuous domain $\mathcal{H}$. This section documents these approaches, their theoretical foundations, and the computational barriers that led us to pursue the convex alternative. We present this analysis to contextualize our methodological choices and guide future research.

F.1. Approach 1: Direct Squashing and Non-Convexity

A common solution for ensuring global constraint satisfaction is to apply a squashing function $\sigma : \mathbb{R} \to$ [7] to an unconstrained function, for example, the logistic function $\sigma(z) = 1/(1 + e^{-z})$. The fundamental issue is that the composition of the logarithm with the mixture under squashing renders the objective non-convex. To see this explicitly, consider the data term for a single observation:

$$\ell(g_i) = -\log\left[\sigma(g_i)f_0(p_i) + (1 - \sigma(g_i))f_1(p_i)\right] \tag{56}$$

The second derivative with respect to the latent value g_i is:

$$\frac{\partial^2 \ell}{\partial g_i^2} = \frac{\partial}{\partial g_i}\left[\frac{-\sigma'(g_i)(f_0(p_i) - f_1(p_i))}{\sigma(g_i)f_0(p_i) + (1-\sigma(g_i))f_1(p_i)}\right] \tag{57}$$

$$= \frac{-\sigma''(g_i)(f_0 - f_1)(\sigma f_0 + (1-\sigma)f_1) + (\sigma'(g_i))^2(f_0 - f_1)^2}{[\sigma(g_i)f_0(p_i) + (1-\sigma(g_i))f_1(p_i)]^2} \tag{58}$$

For the logistic function, the second derivative $\sigma''(z) = \sigma(z)(1-\sigma(z))(1-2\sigma(z))$ changes sign depending on whether $\sigma(z)$ is above or below 0.5. This means the loss function is neither convex nor concave in g_i, even for a single data point. The global objective, being a sum over N such terms plus a convex regularizer, inherits this non-convexity.

The non-convex landscape leads to multiple stationary points, many of which are local minima. Different initializations of $\mathbf{c}$ can converge to qualitatively different solutions with vastly different objective values. Moreover, the objective surface often exhibits regions where the gradient is near-zero but the Hessian has both positive and negative eigenvalues, causing optimization algorithms to stall. In our experiments, we found that the direct squashing approach failed to converge within reasonable iteration budgets, oscillating between different regions of parameter space. Even when optimization converged, the resulting $\alpha(\text{loc})$ functions often exhibited pathological behavior such as rapid oscillations between extreme values (near 0 and 1) in regions with sparse data.

F.2. Approach 2: The Semi-Infinite Programming (SIP) Formulation

Enforcing $\alpha(x) \in [0,1]$ for all $x \in \mathcal{H}$ is a **semi-infinite programming (SIP)** problem (**??**), an optimization with finite decision variables but infinitely many constraints:

$$\begin{aligned} \min_{\mathbf{c}\in\mathbb{R}^N} \quad & -\sum_{i=1}^{N} \log\left[\alpha(x_i)f_0(p_i) + (1-\alpha(x_i))f_1(p_i)\right] + \lambda \mathbf{c}^T K \mathbf{c} \\ \text{s.t.} \quad & 0 \le \alpha(x) \le 1, \quad \forall x \in \mathcal{H} \end{aligned} \tag{59}$$

where $\alpha(x) = \sum_{j=1}^{N} c_j K(x, x_j)$.

Classical SIP methods include: (1) discretization with adaptive refinement, (2) exchange methods that iteratively add violated constraints, (3) reduction to finite equivalent constraints via problem structure, and (4) barrier/penalty methods with sampling-based approximation (**?**). Here, each method corresponds to a classical SIP approach: local reduction uses KKT finite reduction (**?**), polynomial SDP applies moment relaxations (**?**), and the barrier method employs interior point penalties (**?**). However, SIP theory assumes convex objectives and our non-convex mixture likelihood eliminates convergence guarantees, and constraint violation patterns depend on kernel choice in complex ways.

Method 1: Local Reduction via Critical Points. Extreme constraint violations can only occur at critical points in the interior or at domain boundaries. For upper boundary violations ($\alpha(x) > 1$), critical points satisfy:

$$\alpha(x) = \sum_{j=1}^{N} c_j K(x, x_j) = 1 \tag{60}$$

$$\nabla_x \alpha(x) = \sum_{j=1}^{N} c_j \nabla_x K(x, x_j) = 0 \tag{61}$$

$$\nabla^2_{xx} \alpha(x) \preceq 0 \tag{62}$$

For Gaussian RBF kernels, this yields a system of $d+1$ nonlinear equations in d unknowns, solvable via Newton-Raphson when well-conditioned. By Bézout's theorem, a system of d polynomial equations of degree $\mathcal{O}(d)$ admits $\mathcal{O}(2^d d^d)$ real solutions. Cost per critical point: $\mathcal{O}(d^3)$ per Newton iteration gives a total cost of:

$$\text{Cost}_{\text{local}} = \mathcal{O}(2^d d^{d+3} N) \tag{63}$$

So, while under compactness of $\mathcal{H}$ and non-degeneracy of K, all constraint-violating critical points are identified in finite time, the solution is intractable for large d due to exponential scaling and ill-conditioning. Moreover, Jacobian conditioning deteriorates as $\kappa(J) \sim \mathcal{O}(N^{d/(d+2)})$, causing numerical instability. In practice, we tried few common solvers, all weren't able to solve the set of equations fro $d \geq 10$.

Method 2: Polynomial SDP Relaxation. Approximate the non-polynomial mixture likelihood via Taylor expansion, then apply semidefinite programming relaxations for global certificates. Expand $\log(h_i(\alpha))$ around $\alpha = 0.5$ with $\beta_i = \alpha(x_i) - 0.5$ and mixing ratio $r_i = (f_0(p_i) - f_1(p_i))/m_i$:

$$\log(h_i) \approx \log(m_i) + \beta_i r_i - \frac{(\beta_i r_i)^2}{2} + \frac{(\beta_i r_i)^3}{3} - \frac{(\beta_i r_i)^4}{4} \tag{64}$$

Substituting $\beta_i = \sum_j c_j K(x_i, x_j) - 0.5$ yields a degree-4 polynomial in $\mathbf{c}$. The Lasserre SDP hierarchy constructs moment matrices $M_k(y)$ of dimension $\binom{N+k}{k}$ that provide increasingly tight convex relaxations.

For bounded densities with $\max_i |f_0(p_i)|, |f_1(p_i)| \leq B$, Taylor error is $\mathcal{O}(|\beta_i|^5)$. As relaxation order $k \to \infty$, SDP converges to the global polynomial optimum. Nevertheless, the computational memory requirements for moment matrix:

$$\text{Memory} = 8 \cdot \frac{\binom{N+k}{k}(\binom{N+k}{k} + 1)}{2} \text{ bytes} \tag{65}$$

Concrete limits:

$$\begin{aligned} N = 50, k = 2 :&\quad 7.03 \text{ MB} \\ N = 100, k = 2 :&\quad 106 \text{ MB} \\ N = 50, k = 4 :&\quad 400 \text{ GB} \end{aligned}$$

Making this solution non-practical.

F.3. Approach 3: Barrier Method with Tail-Aware Sampling

Finally, here we describe the barrier method for which we compare in the evaluations section. Transform the semi-infinite constraint set into a penalized objective using logarithmic barrier functions, then approximate the resulting domain integral via importance sampling that concentrates samples in regions most likely to violate constraints. The classical barrier method for constrained optimization replaces hard constraints with smooth penalty terms that approach infinity at the boundary. For the semi-infinite programming problem:

$$\begin{aligned} \min_{\mathbf{c} \in \mathbb{R}^N} \quad & -\sum_{i=1}^{N} \log\left[\alpha(x_i) f_0(p_i) + (1 - \alpha(x_i)) f_1(p_i)\right] + \lambda \mathbf{c}^T K \mathbf{c} \\ \text{s.t.} \quad & 0 \leq \alpha(x) \leq 1, \quad \forall x \in \mathcal{H} \end{aligned} \tag{66}$$

we reformulate as:

$$\mathcal{L}_\nu(\mathbf{c}) = -\sum_{i=1}^{N} \log(h_i(\alpha(x_i))) + \lambda \mathbf{c}^T K \mathbf{c} - \frac{1}{\nu} \int_{\mathcal{H}} [\log(\alpha(x)) + \log(1 - \alpha(x))]\, dx \tag{67}$$

where $\nu > 0$ is the barrier parameter controlling penalty strength and the integral is over the Lebesgue measure on $\mathcal{H} \subseteq \mathbb{R}^d$. The barrier terms $-\log(\alpha(x))$ and $-\log(1 - \alpha(x))$ create increasingly steep penalties as $\alpha(x)$ approaches 0 or 1, respectively. As $\nu \to \infty$, these penalties force $\alpha(x)$ to remain strictly within $(0, 1)$ throughout the domain. The method proceeds by solving a sequence of problems with increasing ν:

$$\nu_0 = 1, \quad \nu_{k+1} = \beta \nu_k, \quad \beta \in [1.1, 1.5] \tag{68}$$

Starting from small ν_0 (weak constraints) allows easier optimization, while gradually increasing ν tightens constraints. Each iteration warm-starts from the previous solution. The barrier objective contains the integral:

$$I(\mathbf{c}) = \int_{\mathcal{H}} [\log(\alpha(x)) + \log(1 - \alpha(x))]\, dx \tag{69}$$

This integral has **no closed form for general kernels and domains**, with direct numerical quadrature (e.g., Gaussian quadrature grids) becomes intractable in large d .

F.3.1. Monte Carlo Approximation via Importance Sampling.

We approximate the integral using Monte Carlo integration with M samples $\{z_m\}_{m=1}^M$ drawn from a proposal distribution $q(x)$:

$$I(\mathbf{c}) = \int_{\mathcal{H}} \frac{f(x)}{q(x)} q(x)\, dx \approx \frac{1}{M} \sum_{m=1}^{M} \frac{f(z_m)}{q(z_m)} \tag{70}$$

where $f(x) = \log(\alpha(x)) + \log(1 - \alpha(x))$ and $q(x)$ is the sampling density. The key question is: *what distribution q minimizes variance and ensures constraint violation detection?* For that, we construct a kernel density estimate from the N observed data locations:

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} K_{\text{KDE}}(x, x_i) \tag{71}$$

where $K_{\text{KDE}}(x, y) = \frac{1}{h^d} K_0\left(\frac{x-y}{h}\right)$ is a probability kernel (typically Gaussian) with bandwidth parameter $h > 0$. The bandwidth h critically determines sampling quality. We employ two standard methods: *1. Scott's Rule*:

$$h_{\text{Scott}} = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} N^{-\frac{1}{d+4}} \hat{\sigma} \tag{72}$$

where $\hat{\sigma}$ is the empirical standard deviation of the data locations (computed per-dimension and averaged). *2. Cross-Validation* (optimal, expensive):

$$h_{\text{CV}} =_h \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_{-i}(x_i; h) - \delta(x_i))^2 \tag{73}$$

where $\hat{p}_{-i}(x; h)$ is the leave-one-out KDE excluding point x_i, and δ is the Dirac delta. In practice, this minimizes integrated squared error via grid search over candidate h values.

The Critical Observation is that naive sampling from $\hat{p}(x)$ concentrates samples where data is dense. However, constraint violations $\alpha(x) \notin [0, 1]$ may occur most violently in *tail regions* far from training data, where RKHS extrapolation becomes unreliable. Sampling from $\hat{p}(x)$ thus *misses exactly the regions we need to monitor*. To address this, we design a hybrid distribution that balances three competing objectives:

$$p_{\text{hybrid}}(x) = \rho_1 \hat{p}(x) + \rho_2 q_{\text{tail}}(x) + \rho_3 u(x) \tag{74}$$

where: **Data-Dense Zone** ($\rho_1 = 0.3$) are sampled from KDE $\hat{p}(x)$. These regions contribute most to the data-fit term in $\mathcal{L}_\nu$. **Tail Zone** ($\rho_2 = 0.5$) are samples from inverse density:

$$q_{\text{tail}}(x) = \frac{w(x)}{\int_{\mathcal{H}} w(y) dy}, \quad w(x) = \frac{1}{\hat{p}(x) + \epsilon} \tag{75}$$

where $\epsilon > 0$ (typically $\epsilon = 10^{-6}$) prevents division by zero. These are points where violent violations occur, for which we use rejection sampling with acceptance probability $\propto (\hat{p}(x) + \epsilon)^{-1}$. **Uniform Zone** ($\rho_3 = 0.2$) are sampled uniformly

$$u(x) = \frac{1}{\text{Vol}(\mathcal{H})} \tag{76}$$

Finally, the choice $(\rho_1, \rho_2, \rho_3) = (0.3, 0.5, 0.2)$ comes from hyperparameter optimization

F.4. Complete Gradient Computation

The gradient of $\mathcal{L}_\nu$ with respect to coefficient c_k has three terms:

1. Data Term (exact):

$$\frac{\partial}{\partial c_k} \left(-\sum_{i=1}^{N} \log(h_i) \right) = -\sum_{i=1}^{N} \frac{1}{h_i(\alpha(x_i))} \frac{\partial h_i}{\partial c_k} \tag{77}$$

where:

$$\begin{aligned}\frac{\partial h_i}{\partial c_k} &= \frac{\partial}{\partial c_k}\left[\alpha(x_i)f_0(p_i) + (1-\alpha(x_i))f_1(p_i)\right] && (78)\\ &= \frac{\partial \alpha(x_i)}{\partial c_k}(f_0(p_i) - f_1(p_i)) && (79)\\ &= K(x_i, x_k)(f_0(p_i) - f_1(p_i)) && (80)\end{aligned}$$

Thus:

$$\frac{\partial}{\partial c_k}(\text{Data}) = -\sum_{i=1}^{N} \frac{K(x_i, x_k)(f_0(p_i) - f_1(p_i))}{\alpha(x_i)f_0(p_i) + (1-\alpha(x_i))f_1(p_i)} \tag{81}$$

2. Regularization Term (exact):

$$\frac{\partial}{\partial c_k}\left(\lambda \mathbf{c}^T K \mathbf{c}\right) = 2\lambda \sum_{j=1}^{N} c_j K(x_j, x_k) \tag{82}$$

3. Barrier Term (Monte Carlo approximation):

The exact gradient is:

$$\frac{\partial}{\partial c_k}\left(-\frac{1}{\nu}\int_{\mathcal{H}}[\log(\alpha(x)) + \log(1-\alpha(x))]dx\right) = -\frac{1}{\nu}\int_{\mathcal{H}} K(x, x_k)\left[\frac{1}{\alpha(x)} - \frac{1}{1-\alpha(x)}\right]dx \tag{83}$$

Using importance sampling with $\{z_m\}_{m=1}^{M} \sim p_{\text{hybrid}}$:

$$\frac{\partial}{\partial c_k}(\text{Barrier}) \approx -\frac{\text{Vol}(\mathcal{H})}{\nu M}\sum_{m=1}^{M} \frac{K(z_m, x_k)}{p_{\text{hybrid}}(z_m)}\left[\frac{1}{\alpha(z_m)} - \frac{1}{1-\alpha(z_m)}\right] \tag{84}$$

where $\text{Vol}(\mathcal{H})$ is the domain volume (e.g., for $\mathcal{H} = [a, b]^d$, $\text{Vol} = (b-a)^d$).

Complete Gradient:

$$\begin{aligned}\frac{\partial \mathcal{L}_\nu}{\partial c_k} \approx -\sum_{i=1}^{N} \frac{K(x_i, x_k)(f_0(p_i) - f_1(p_i))}{h_i(\alpha(x_i))} + 2\lambda \sum_{j=1}^{N} c_j K(x_j, x_k) && (85)\\ -\frac{\text{Vol}(\mathcal{H})}{\nu M}\sum_{m=1}^{M} \frac{K(z_m, x_k)}{p_{\text{hybrid}}(z_m)}\left[\frac{1}{\alpha(z_m)} - \frac{1}{1-\alpha(z_m)}\right] && (86)\end{aligned}$$

Complete Algorithm:

Algorithm 3 KDE-Guided Tail-Aware Barrier Method

```
Require: Training data $\{x_i, p_i\}_{i=1}^N$, kernel $K$, densities $f_0, f_1$, regularization $\lambda$
Ensure: Optimal coefficients $\mathbf{c}^*$
Initialization:
  Set $\mathbf{c}^0 = \mathbf{0}$ (or solve unconstrained problem)
  Set $\nu_0 = 1.0$, $\beta = 1.2$, learning rate $\eta_0 = 0.01$
  Choose domain bounds $\mathcal{H}$ (e.g., bounding box of data + margin)
  Compute Vol$(\mathcal{H})$

KDE Construction:
  Select bandwidth $h$ via Scott's rule or cross-validation
  Define $\hat{p}(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^d} \exp\left(-\frac{\|x-x_i\|^2}{2h^2}\right)$ (Gaussian KDE)
  Set $\epsilon = 10^{-6}$

Barrier Path-Following:
for barrier iteration $k = 0, 1, \ldots, K_{\max}$ (typically $K_{\max} = 50$) do

     Gradient Descent Loop:
   for inner iteration $t = 0, 1, \ldots, T_{\max}$ (typically $T_{\max} = 100$) do

         // Generate hybrid samples
         $\mathcal{S}_1 \leftarrow$ Sample $M_1 = \lfloor \rho_1 M \rfloor$ points from $\hat{p}(x)$
         $\mathcal{S}_2 \leftarrow$ Sample $M_2 = \lfloor \rho_2 M \rfloor$ points from $q_{\text{tail}}(x) \propto (\hat{p}(x) + \epsilon)^{-1}$
         $\mathcal{S}_3 \leftarrow$ Sample $M_3 = M - M_1 - M_2$ points from Uniform$(\mathcal{H})$
         $\{z_m\}_{m=1}^M \leftarrow \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$

         // Evaluate $\alpha$ and $h_i$ at all sample points
      for $m = 1, \ldots, M$ do
            $\alpha(z_m) \leftarrow \sum_{j=1}^N c_j^{(k,t)} K(z_m, x_j)$
            $p_{\text{hybrid}}(z_m) \leftarrow \rho_1 \hat{p}(z_m) + \rho_2 (\hat{p}(z_m) + \epsilon)^{-1}/Z + \rho_3/\text{Vol}(\mathcal{H})$
      end for
      for $i = 1, \ldots, N$ do
            $\alpha(x_i) \leftarrow \sum_{j=1}^N c_j^{(k,t)} K(x_i, x_j)$
            $h_i \leftarrow \alpha(x_i) f_0(p_i) + (1 - \alpha(x_i)) f_1(p_i)$
      end for

         // Compute gradient for each coefficient
      for $k' = 1, \ldots, N$ do
            // Data term
            $g_{\text{data}} \leftarrow -\sum_{i=1}^N \frac{K(x_i, x_{k'})(f_0(p_i) - f_1(p_i))}{h_i}$

            // Regularization term
            $g_{\text{reg}} \leftarrow 2\lambda \sum_{j=1}^N c_j^{(k,t)} K(x_j, x_{k'})$

            // Barrier term (Monte Carlo)
            $g_{\text{barrier}} \leftarrow -\frac{\text{Vol}(\mathcal{H})}{\nu_k M} \sum_{m=1}^M \frac{K(z_m, x_{k'})}{p_{\text{hybrid}}(z_m)} \left[\frac{1}{\alpha(z_m)} - \frac{1}{1-\alpha(z_m)}\right]$

            $\frac{\partial \mathcal{L}_{\nu_k}}{\partial c_{k'}} \leftarrow g_{\text{data}} + g_{\text{reg}} + g_{\text{barrier}}$
      end for

         // Gradient descent update
         $\mathbf{c}^{(k,t+1)} \leftarrow \mathbf{c}^{(k,t)} - \eta_t \nabla \mathcal{L}_{\nu_k}(\mathbf{c}^{(k,t)})$

         // Check convergence
      if $\|\nabla \mathcal{L}_{\nu_k}(\mathbf{c}^{(k,t+1)})\| < \text{tol}_{\text{inner}}$ then
            break
      end if
   end for

```

F.4.1. Practical Limitations

Despite strong theoretical foundations and favorable asymptotic complexity, the barrier method proved impractical in our experiments:

1. **Sampling Sensitivity:** The mixing weights (ρ_1, ρ_2, ρ_3) require problem-specific tuning. Too much tail allocation ($\rho_2 > 0.6$) introduces excessive gradient noise; too little ($\rho_2 < 0.3$) misses violations.

2. **Bandwidth Selection:** KDE bandwidth h critically affects tail sampling quality. Scott's rule often oversmooths for $d \geq 5$; cross-validation is expensive and can be unstable.

3. **Barrier Schedule:** The growth rate β and starting value ν_0 require careful tuning. Aggressive schedules ($\beta > 1.5$) cause convergence failure; conservative schedules ($\beta < 1.1$) waste computation.

4. **Gradient Variance:** Even with $M = 10^4$ samples, Monte Carlo variance in the barrier gradient necessitates small learning rates ($\eta \sim 10^{-3}$), requiring hundreds of iterations per barrier level.

5. **Non-Convexity Persists:** The method still optimizes the non-convex mixture likelihood, inheriting all local minima issues. Multiple random initializations are required, multiplying computational cost.

6. **Domain Specification:** Requires explicit domain bounds $\mathcal{H}$. Infinite or unbounded domains must be truncated, potentially missing distant violations.

7. **Numerical Instability:** Near constraint boundaries ($\alpha \approx 0$ or $\alpha \approx 1$), the terms $1/\alpha$ and $1/(1-\alpha)$ become numerically unstable, requiring careful regularization ($\alpha \leftarrow \text{clip}(\alpha, 10^{-8}, 1 - 10^{-8})$).