Supplementary information

Contents

1 Previous studies on edge superelevation

Several previous studies have investigated the formation of edge profiles in thin films under various process conditions and have explained the underlying physical principles. Dobroth and Erwin [1] discussed edge beads in cast polymer films and identified surface tension, die swell, and edge stress effects as the main causes. In particular, they emphasized that the edge stress effect induced by film

stretching is the predominant cause. Gutoff et al. [2] addressed heavy edges in premetered coating and speculated that film stretching would be the common cause in slot-die coating. Schmitt et al. [3] developed a technique to quantitatively measure edge profiles in slot-die coating and proposed shape metrics to describe the edge elevation effect. Bitsch et al. [4] showed that increasing the low-shear viscosity of slurries suppresses the edge elevation effect. Schmitt [5] extended Dobroth's film stretching model by incorporating adhesion between the substrate and coating layer, thereby proposing a film-stretching model for slot-die coating. Spiegel et al. [6] investigated the effects of coating speed, dimensionless gap, and die geometry on edge profiles and proposed three types of edge shapes. Spiegel et al. [7] further examined the influence of coating gap and angle of attack on edge formation and proposed an optimized setup for ultrathick electrodes. Yoon et al. [8] collected a large dataset of edge profiles across various process conditions, including slurry properties, coating speed, feed flow rate, and slot-die configurations. They also refined the definitions of shape metrics to enable their application to non-canonical profiles, and through significance analysis revealed that edge profile formation primarily depends on the gap-to-thickness ratio R_{gt} as well as the capillary number Ca.

Collectively, these studies characterize the relationship between process conditions and edge morphology. The shared engineering goal is to identify quality windows, which are the regions of process space in which edge shape reliably meets quality criteria. This concept extends the classical operability-window framework for coating-bead stability [9] to the stricter requirement of shape quality. However, prior work has largely treated the quality window as a conceptual target, identifying candidate regions through visual inspection of observed profiles or threshold-based heuristics applied to shape metrics. A probabilistic formulation that quantifies the likelihood of meeting quality criteria across process space, accounting for the variability inherent in manufacturing, has not been established before the present work.

2 Regression models

2.1 Quantile regression

Let

$$m = g(X) + \varepsilon \tag{1}$$

be a shape metric, where ε is a random noise term. We are interested in estimating the τ-th quantile $Q_\tau(x)$ of $m \mid X = x$ from observed data. There are two approaches to estimating $Q_\tau(x)$: the cumulative distribution function (CDF) approach and the estimating equation (EE) approach [10].

The CDF approach estimates the entire conditional CDF of $m \mid X = x$ and then inverts it to obtain the quantile. The most common CDF approach assumes a parametric distribution model for $m \mid X = x$ and estimates its parameters. Examples include prediction intervals in ordinary least-squares regression

and posterior distributions in Gaussian process regression. Alternatively, a non-parametric distribution model can be assumed, such as the Dirichlet process mixture model [11]. It is also possible to directly obtain the empirical distribution of $m \mid X = x$ without imposing distributional assumptions, as in quantile regression forests [12]. The advantage of the CDF approach is that once the distribution is estimated, any quantile can be obtained directly. The disadvantage is that it is inefficient if only a small number of quantiles are of interest, as estimating the entire distribution may require more data and computational resources.

The EE approach directly estimates $Q_\tau(x)$ by regression using a *τ-quantile loss* $\mathcal{L}_\tau$, without estimating the entire distribution [13]. The model is defined as

$$Q_\tau(x) = \arg\min_q E\left[\mathcal{L}_\tau(m, q) \mid X = x\right], \quad \mathcal{L}_\tau(m, q) = \begin{cases} \tau|m - q|, & m > q, \\ (1-\tau)|m - q|, & m \le q. \end{cases} \tag{2}$$

The CDF can also be constructed from this approach by solving Eq. 2 for multiple τ values and interpolating or smoothing the discrete quantiles [14]. The advantage of the EE approach is that it can be directly plugged into any regression model. Examples include linear regression [15, 16], kernel regression [17], Gaussian process [10, 18], tree-based regression [19], and neural networks [20,21]. Additionally, the EE approach requires fewer assumptions than the CDF approach. The disadvantage is that when multiple quantiles are of interest, separate models need to be trained. This may lead to a phenomenon called *quantile crossing*, where a lower quantile is estimated to be greater than a higher quantile at some x.

Quantile regression by Eq. 2 can be extended to the Bayesian framework by using asymmetric Laplace distribution (ALD) likelihood [16]. For simplicity, we consider a linear quantile regression model $Q_\tau(x) = \boldsymbol{x}^\top \boldsymbol{\beta}_\tau$, where $\boldsymbol{x}$ is the design vector depending on x and $\boldsymbol{\beta}_\tau$ is the vector of regression coefficients. By Eq. 2, the $\boldsymbol{\beta}_\tau$ can be estimated by

$$\hat{\boldsymbol{\beta}}_\tau = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \mathcal{L}_\tau(m_i, \boldsymbol{x}_i^\top \boldsymbol{\beta}), \tag{3}$$

For observations $\mathbf{m} = (m_1, \ldots, m_n)$, this is equivalent to placing a likelihood

$$\mathbb{P}\left(\mathbf{m} \mid \boldsymbol{\beta}\right) = \prod_{i=1}^{n} \tau(1-\tau) \exp\left\{-\mathcal{L}_\tau(m_i, \boldsymbol{x}_i^\top \boldsymbol{\beta})\right\} \tag{4}$$

and maximizing it [16]. The posterior distribution of $\boldsymbol{\beta}$ is given by

$$\mathbb{P}\left(\boldsymbol{\beta} \mid \mathbf{m}\right) \propto \mathbb{P}\left(\mathbf{m} \mid \boldsymbol{\beta}\right) \mathbb{P}\left(\boldsymbol{\beta}\right), \tag{5}$$

where $\mathbb{P}\left(\boldsymbol{\beta}\right)$ is the prior density of $\boldsymbol{\beta}$. If there is no prior knowledge, an improper uniform prior can be used for $\mathbb{P}\left(\boldsymbol{\beta}\right)$. The distribution of $Q_\tau(x_*)$ at a candidate point x_* can be obtained by sampling methods such as Markov chain Monte

Carlo (MCMC). It was also reported that by converting the ALD likelihood to location-scale mixture of normal distributions, the sampling can be further simplified using Gibbs sampling [22].

2.2 Gaussian process regression

The Gaussian process (GP) regression model assumes that the latent function $f(X)$ follows a specific GP prior [23]. The prior defines what the expected value of $f(X)$ at a given $X \in \mathbb{R}^d$ is, and how the values of $f(X)$ at different X are correlated. The prior for f is given by

$$f \sim \mathcal{GP}\left(\mu, k\right), \tag{6}$$

where $\mu(\cdot)$ is the mean function and $k(\cdot,\cdot)$ is a covariance kernel function. This means that for a given set of points $\mathbf{X}$, the prior distribution of $f(\mathbf{X})$ is

$$f(\mathbf{X}) \sim \mathcal{N}\left(\mu(\mathbf{X}), k(\mathbf{X},\mathbf{X})\right). \tag{7}$$

Combining Eqs. 54 and 7, the prior distribution of the observed $\boldsymbol{\ell}$ is given by

$$\boldsymbol{\ell} \sim \mathcal{N}\left(\mu(\mathbf{X}), k(\mathbf{X},\mathbf{X}) + \sigma^2_{\text{noise}}\mathbf{I}\right), \tag{8}$$

where $\mathbf{I}$ is the identity matrix. Note that, if f is fixed, then Eq. 8 is reduced to the likelihood

$$\boldsymbol{\ell} \mid f \sim \mathcal{N}\left(f(\mathbf{X}), \sigma^2_{\text{noise}}\mathbf{I}\right). \tag{9}$$

Let $\mathcal{D}_n = (\mathbf{X}_n, \boldsymbol{\ell}_n)$ be the observed dataset with size n, where $\mathbf{X}_n \in \mathbb{R}^{n\times d}$ and $\boldsymbol{\ell}_n \in \mathbb{R}^{n\times 1}$. Consider a candidate point X_*. By Eqs. 7 and 8, the joint distribution of $\boldsymbol{\ell}_n$ and $f(X_*)$ is

$$\begin{bmatrix} \boldsymbol{\ell}_n \\ f(X_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_n \\ \mu(X_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_n + \sigma^2_{\text{noise}}\mathbf{I}_n & \mathbf{k}_{n*} \\ \mathbf{k}_{n*}^\top & k(X_*, X_*) \end{bmatrix}\right), \tag{10}$$

where $\boldsymbol{\mu}_n = \mu(\mathbf{X}_n)$, $\mathbf{k}_{n*} = k(\mathbf{X}_n, X_*)$, $\mathbf{K}_n = k\left(\mathbf{X}_n, \mathbf{X}_n\right)$, and $\mathbf{I}_n$ is the identity matrix of size n. By conditioning Eq. 10, the posterior distribution of $f(X_*)$ is given by

$$f(X_*) \mid \boldsymbol{\ell}_n \sim \mathcal{N}\big(\mu_*(X_*), \sigma^2_*(X_*)\big), \tag{11}$$

where

$$\mu_*(X_*) = \mu(X_*) + \mathbf{k}_{n*}\left[\mathbf{K}_n + \sigma^2_{\text{noise}}\mathbf{I}\right]^{-1}\left(\boldsymbol{\ell}_n - \boldsymbol{\mu}_n\right), \tag{12}$$

$$\sigma^2_*(X_*) = k(X_*, X_*) - \mathbf{k}_{n*}\left[\mathbf{K}_n + \sigma^2_{\text{noise}}\mathbf{I}\right]^{-1}\mathbf{k}_{n*}^\top. \tag{13}$$

In Eqs. 54 and 6, the noise variance σ^2_{noise}, the mean μ and the covariance k are assumed to be known. In practice, however, these hyperparameters need to be estimated from data. Assume parametric forms for μ and k, and denote the vector of these parameters and σ^2_{noise} as $\boldsymbol{\theta}$. Then, for a fixed $\boldsymbol{\theta}$, the prior distribution of f is given by

$$f \mid \boldsymbol{\theta} \sim \mathcal{GP}\left(\mu_{\boldsymbol{\theta}}, k_{\boldsymbol{\theta}}\right), \tag{14}$$

where $\mu_{\boldsymbol{\theta}}$ and $k_{\boldsymbol{\theta}}$ are the μ and k parameterized by $\boldsymbol{\theta}$, respectively. As in Eq. 7, this leads to

$$f(\mathbf{X}) \mid \boldsymbol{\theta} \sim \mathcal{N}\left(\mu_{\boldsymbol{\theta}}(\mathbf{X}), k_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})\right). \tag{15}$$

With a slight abuse of notation, we use the probability symbol $\mathbb{P}$ to denote the probability measure of a random variable. $\boldsymbol{\theta}$ is estimated by maximizing the posterior density $\mathbb{P}\left(\boldsymbol{\theta} \mid \boldsymbol{\ell}_n\right)$. By Bayes' theorem, $\mathbb{P}\left(\boldsymbol{\theta} \mid \boldsymbol{\ell}_n\right) \propto \mathbb{P}\left(\boldsymbol{\ell}_n \mid \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta}\right)$. The distribution of $\boldsymbol{\ell}_n \mid \boldsymbol{\theta}$ is obtained by marginalizing Eq. 9 over f as

$$\mathbb{P}\left(\boldsymbol{\ell}_n \mid \boldsymbol{\theta}\right) = \int \mathbb{P}\left(\boldsymbol{\ell}_n \mid f\right) \mathbb{P}\left(f \mid \boldsymbol{\theta}\right) \mathrm{d}f, \tag{16}$$

which, by Eqs. 9 and 14, leads to

$$\boldsymbol{\ell}_n \mid \boldsymbol{\theta} \sim \mathcal{N}\left(\mu_{\boldsymbol{\theta}}(\mathbf{X}_n), k_{\boldsymbol{\theta}}(\mathbf{X}_n, \mathbf{X}_n) + \sigma^2_{\text{noise}}\mathbf{I}\right), \tag{17}$$

and thus

$$\mathbb{P}\left(\boldsymbol{\ell}_n \mid \boldsymbol{\theta}\right) = \frac{1}{(2\pi)^{n/2}\boldsymbol{\Sigma}^{1/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{\ell}_n - \mu_{\boldsymbol{\theta}}(\mathbf{X}_n)\right)^\top \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\ell}_n - \mu_{\boldsymbol{\theta}}(\mathbf{X}_n)\right)\right), \tag{18}$$

where $\boldsymbol{\Sigma} = k_{\boldsymbol{\theta}}(\mathbf{X}_n, \mathbf{X}_n) + \sigma^2_{\text{noise}}\mathbf{I}$. The hyperparameters $\boldsymbol{\theta}$ and σ^2_{noise} are estimated by maximizing either the log marginal likelihood $\log \mathbb{P}\left(\boldsymbol{\ell}_n \mid \boldsymbol{\theta}\right)$ or assuming a prior $\mathbb{P}(\boldsymbol{\theta})$ and maximizing the log posterior $\log \mathbb{P}\left(\boldsymbol{\ell}_n \mid \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta}\right)$. If the lengthscales of the kernel are anisotropic, estimating them in this manner is referred to as automatic relevance determination (ARD) [24].

Scaling the data and choosing an appropriate prior are important considerations for Gaussian process modeling. For numerical stability, it is customary to apply min-max scaling to each dimension of X using the bounds of the design space. For the observed response $\boldsymbol{\ell}_n$, standardization to zero mean and unit variance is recommended. These scaling procedures are implicitly adopted in our implementation. The choice of μ represents the prior belief about the expected value of f, such as a physical model of the system. When no prior knowledge is available, $\mu(X)$ is usually set to a constant for all X, e.g., zero for standardized $\boldsymbol{\ell}_n$. The choice of k dictates the shape of f, such as its smoothness and periodicity. Matérn kernels are commonly employed for non-periodic functions.

2.3 Gaussian process quantile regression

For nonparametric Bayesian estimation, quantile functions can be modeled using Gaussian process regression. Let $\mathbf{X}$ and $\mathbf{m}$ be the observed input and output data, respectively. As in Eq. 6, we place a GP prior on the quantile function Q_τ as

$$Q_\tau \sim \mathcal{GP}\left(\mu, k\right). \tag{19}$$

Our goal is to estimate $\mathbb{E}\left[Q_\tau(X_*) \mid \mathbf{m}\right]$ which serves as the τ-quantile at a candidate point X_*.

As in Eq. 7, Eq. 19 leads to

$$\boldsymbol{Q}_\tau = Q_\tau(\mathbf{X}) \sim \mathcal{N}\left(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X})\right). \tag{20}$$

The likelihood is given by

$$m_i \mid Q_\tau(x_i) \sim \mathrm{ALD}_\tau\left(\mu = Q_\tau(x_i), \sigma\right), \tag{21}$$

with location μ and scale σ, which correspond to Eq. 9. Assuming independent observations, the likelihood for $\mathbf{m}$ becomes

$$\mathbb{P}\left(\mathbf{m} \mid \boldsymbol{Q}_\tau\right) = \prod_{i=1}^{n} \frac{\tau(1-\tau)}{\sigma} \exp\left\{-\frac{1}{\sigma}\mathcal{L}_\tau(m_i, Q_\tau(x_i))\right\}, \tag{22}$$

as in Eq. 4. Unlike Eq. 11, the posterior distribution at a candidate point $Q_\tau(X_*) \mid \mathbf{m}$ has no closed-form solution. Instead, its density is given by marginalization

$$\mathbb{P}\left(Q_\tau(X_*) \mid \mathbf{m}\right) = \int \mathbb{P}\left(Q_\tau(X_*) \mid \boldsymbol{Q}_\tau\right) \mathbb{P}\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right) \mathrm{d}\boldsymbol{Q}_\tau. \tag{23}$$

The first term, $Q_\tau(X_*) \mid \boldsymbol{Q}_\tau$, is a conditional Gaussian because of Eq. 20. By Bayes' theorem, the second term is given by $\mathbb{P}\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right) \propto \mathbb{P}\left(\mathbf{m} \mid \boldsymbol{Q}_\tau\right) \mathbb{P}\left(\boldsymbol{Q}_\tau\right)$, which is not Gaussian because of Eq. 22. Hence, Eq. 23 has no closed-form solution and needs to be approximated. Numerical methods, such as MCMC, are not scalable for GP with large datasets because of the high dimensionality of $\boldsymbol{Q}_\tau$. Instead, we approximate $\boldsymbol{Q}_\tau \mid \mathbf{m}$ by a Gaussian distribution to acquire analytical approximation of Eq. 23. Previous studies have proposed using expectation propagation (EP) or variational inference (VI) for this purpose [10, 25]. In this work, we use the VI approach, as it is computationally efficient and readily available by software packages. Although the previous work uses the scale-mixture representation of the ALD for VI, we directly use the ALD likelihood in Eq. 22 for simplicity.

Let $p(\cdot)$ denote the true density of a random variable. Our goal is to approximate $p\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right)$ by a density $q(\boldsymbol{Q}_\tau)$ within a family of joint Gaussian distributions $\mathfrak{D}$, such that

$$q^*(\boldsymbol{Q}_\tau) = \underset{q(\boldsymbol{Q}_\tau) \in \mathfrak{D}}{\arg\min}\, \mathrm{KL}\left(q(\boldsymbol{Q}_\tau) \parallel p\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right)\right), \tag{24}$$

where

$$\begin{aligned}
\mathrm{KL}\left(q(\boldsymbol{Q}_\tau) \parallel p\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right)\right) &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log q(\boldsymbol{Q}_\tau)\right] - \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\boldsymbol{Q}_\tau \mid \mathbf{m}\right)\right] \\
&= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log q(\boldsymbol{Q}_\tau)\right] - \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\boldsymbol{Q}_\tau, \mathbf{m}\right)\right] \\
&\quad + \log p(\mathbf{m}).
\end{aligned}$$

Because $\log p(\mathbf{m})$ does not depend on $q(\boldsymbol{Q}_\tau)$, minimizing the KL divergence is

equivalent to maximizing the evidence lower bound (ELBO):

$$\begin{aligned}\mathrm{ELBO}(q) &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\boldsymbol{Q}_\tau, \mathbf{m}\right)\right] - \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log q(\boldsymbol{Q}_\tau)\right] && (25)\\ &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau\right)\right] + \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\boldsymbol{Q}_\tau\right)\right] - \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log q(\boldsymbol{Q}_\tau)\right] && (26)\\ &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau\right)\right] - \mathrm{KL}\left(q(\boldsymbol{Q}_\tau) \parallel p\left(\boldsymbol{Q}_\tau\right)\right). && (27)\end{aligned}$$

Because $q(\boldsymbol{Q}_\tau)$ and $p(\boldsymbol{Q}_\tau)$ are normal distributions, the KL divergence term can be computed in closed form. By Eq. 22, the expected log likelihood term transforms to

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau\right)\right] &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\log \prod_i^n p\left(m_i \mid Q_\tau(x_i)\right)\right] && (28)\\ &= \mathbb{E}_{q(\boldsymbol{Q}_\tau)}\left[\sum_i^n \log p\left(m_i \mid Q_\tau(x_i)\right)\right] && (29)\\ &= \sum_i^n \mathbb{E}_{q(Q_\tau(x_i))}\left[\log p\left(m_i \mid Q_\tau(x_i)\right)\right], && (30)\end{aligned}$$

where $\log p\left(m_i \mid Q_\tau(x_i)\right) = -\mathcal{L}_\tau\left(m_i, Q_\tau(x_i)\right)/\sigma + \text{const}$. Thus, the expected log likelihood can be computed through n one-dimensional numerical integrations, which is computationally feasible. After $q^*(\boldsymbol{Q}_\tau)$ is obtained, Eq. 23 becomes

$$\mathbb{P}\left(Q_\tau(X_*) \mid \mathbf{m}\right) \approx \int \mathbb{P}\left(Q_\tau(X_*) \mid \boldsymbol{Q}_\tau\right) q^*(\boldsymbol{Q}_\tau) \mathrm{d}\boldsymbol{Q}_\tau, \tag{31}$$

which has a closed-form solution, as in Eq. 16.

As in Eq. 14, parametric forms can be assumed for μ and k. Let $\boldsymbol{\theta}$ be the vector of hyperparameters. Then,

$$Q_\tau \mid \boldsymbol{\theta} \sim \mathcal{GP}\left(\mu_{\boldsymbol{\theta}}, k_{\boldsymbol{\theta}}\right), \tag{32}$$

which leads to, as in Eq. 15,

$$\boldsymbol{Q}_\tau \mid \boldsymbol{\theta} \sim \mathcal{N}\left(\mu_{\boldsymbol{\theta}}(\mathbf{X}), k_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X})\right). \tag{33}$$

The posterior distribution of $\boldsymbol{\theta}$ is given by

$$\mathbb{P}\left(\boldsymbol{\theta} \mid \mathbf{m}\right) \propto \mathbb{P}\left(\mathbf{m} \mid \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta}\right), \tag{34}$$

as in Eq. 5. The likelihood $\mathbb{P}\left(\mathbf{m} \mid \boldsymbol{\theta}\right)$ is obtained by

$$\mathbb{P}\left(\mathbf{m} \mid \boldsymbol{\theta}\right) = \int \mathbb{P}\left(\mathbf{m} \mid \boldsymbol{Q}_\tau, \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{Q}_\tau \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{Q}_\tau, \tag{35}$$

as in Eq. 16, and $\boldsymbol{\theta}$ can be estimated either by maximizing its lower bound using ELBO or by using expectation propagation [10, 26]. Estimation by ELBO is as

follows. For any $q(\boldsymbol{Q}_\tau)$, the log likelihood is

$$\log p\left(\mathbf{m} \mid \boldsymbol{\theta}\right) = \log \int q(\boldsymbol{Q}_\tau) \frac{p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau, \boldsymbol{\theta}\right) p\left(\boldsymbol{Q}_\tau \mid \boldsymbol{\theta}\right)}{q(\boldsymbol{Q}_\tau)} \mathrm{d}\boldsymbol{Q}_\tau \tag{36}$$

$$= \log \mathbb{E}_{q(\boldsymbol{Q}_\tau)} \left[\frac{p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau, \boldsymbol{\theta}\right) p\left(\boldsymbol{Q}_\tau \mid \boldsymbol{\theta}\right)}{q(\boldsymbol{Q}_\tau)} \right]. \tag{37}$$

Because log is a concave function, by Jensen's inequality, we have

$$\log p\left(\mathbf{m} \mid \boldsymbol{\theta}\right) \geq \mathbb{E}_{q(\boldsymbol{Q}_\tau)} \left[\log \frac{p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau, \boldsymbol{\theta}\right) p\left(\boldsymbol{Q}_\tau \mid \boldsymbol{\theta}\right)}{q(\boldsymbol{Q}_\tau)} \right] \tag{38}$$

$$= \mathbb{E}_{q(\boldsymbol{Q}_\tau)} \left[\log p\left(\mathbf{m} \mid \boldsymbol{Q}_\tau, \boldsymbol{\theta}\right) \right] - \mathrm{KL}\left(q(\boldsymbol{Q}_\tau) \parallel p\left(\boldsymbol{Q}_\tau \mid \boldsymbol{\theta}\right) \right) \tag{39}$$

$$\tag{40}$$

which is the ELBO. Hence, maximizing the ELBO jointly optimizes $q(\boldsymbol{Q}_\tau)$ and $\boldsymbol{\theta}$.

3 Calibration of classification model

We compare five calibration methods, which differ in the functional form of the score-to-probability mapping and the multiclass decomposition strategy. Among the methods, we choose One-versus-rest sigmoid regression as it achieves the best balance between classification performance and landscape quality of ϕ_{class}.

3.1 Calibration methods

Binary calibration methods such as Platt scaling and isotonic regression are extended to multi-class settings using either the one-versus-rest (OvR) or one-versus-one (OvO) strategy. Note that temperature scaling naturally supports multi-class calibration without decomposition.

Sigmoid OvR (Platt scaling). In the one-versus-rest (OvR) approach, each class k is treated independently. A sigmoid function is fitted to the score $\hat{s}_k$ [27]:

$$q_k(Y) = \frac{1}{1 + \exp\big(a_k \hat{s}_k(Y) + b_k\big)}, \tag{41}$$

where a_k and b_k are learned by minimizing the negative log-likelihood on the calibration (validation) set. The calibrated probability is obtained by normalizing across classes:

$$p_k(Y) = \frac{q_k(Y)}{\sum_{l=1}^{K} q_l(Y)}. \tag{42}$$

Sigmoid OvO. In the one-versus-one (OvO) approach, a sigmoid calibrator is fitted for each class pair (i, j) using only the samples belonging to class i or j. The pairwise probability r_{ij} that the true class is i rather than j is modeled as

$$r_{ij}(Y) = \frac{1}{1 + \exp\big(a_{ij}(\hat{s}_i(Y) - \hat{s}_j(Y)) + b_{ij}\big)}, \tag{43}$$

where a_{ij} and b_{ij} are learned from the binary subproblem. The multiclass probability is obtained by normalized voting [28]:

$$p_k(Y) = \frac{\sum_{j \neq k} r_{kj}(Y)}{\sum_{l=1}^{K} \sum_{j \neq l} r_{lj}(Y)}. \tag{44}$$

Isotonic OvR. Each class k is calibrated using isotonic regression [29], a non-parametric method that fits a monotonically non-decreasing step function g_k to the score–probability relationship:

$$q_k(Y) = g_k\big(\hat{s}_k(Y)\big), \quad \text{subject to} \quad g_k \text{ being non-decreasing}, \tag{45}$$

where g_k is obtained by minimizing $\sum_j \big(g_k(\hat{s}_k(Y_j)) - \mathbf{1}[L_j = k]\big)^2$ under the monotonicity constraint. The calibrated probability is obtained by normalization as $p_k(Y) = q_k(Y) / \sum_{l=1}^{K} q_l(Y)$.

Isotonic OvO. Isotonic OvO follows the same pairwise decomposition as sigmoid OvO, but replaces each sigmoid calibrator with isotonic regression:

$$r_{ij}(Y) = g_{ij}\big(\hat{s}_i(Y) - \hat{s}_j(Y)\big), \quad \text{subject to} \quad g_{ij} \text{ being non-decreasing}, \tag{46}$$

with the same normalized-voting coupling step to obtain multiclass probabilities.

Temperature scaling. Temperature scaling [30] applies a single scalar parameter $\beta > 0$ to all class scores simultaneously:

$$p_k(Y) = \frac{\exp\big(\beta \hat{s}_k(Y)\big)}{\sum_{l=1}^{K} \exp\big(\beta \hat{s}_l(Y)\big)}. \tag{47}$$

The parameter β is optimized by minimizing the negative log-likelihood on the calibration set. Unlike Platt scaling, which fits $2K$ parameters, temperature scaling uses a single parameter, preserving the relative ordering among classes while adjusting the sharpness of the predicted distribution.

3.2 Classification and calibration performance

Figure 1 shows the calibration curves for all five methods across each class. Each curve is constructed by partitioning the predicted probability range $[0, 1]$ into equally spaced bins and plotting the mean predicted probability against

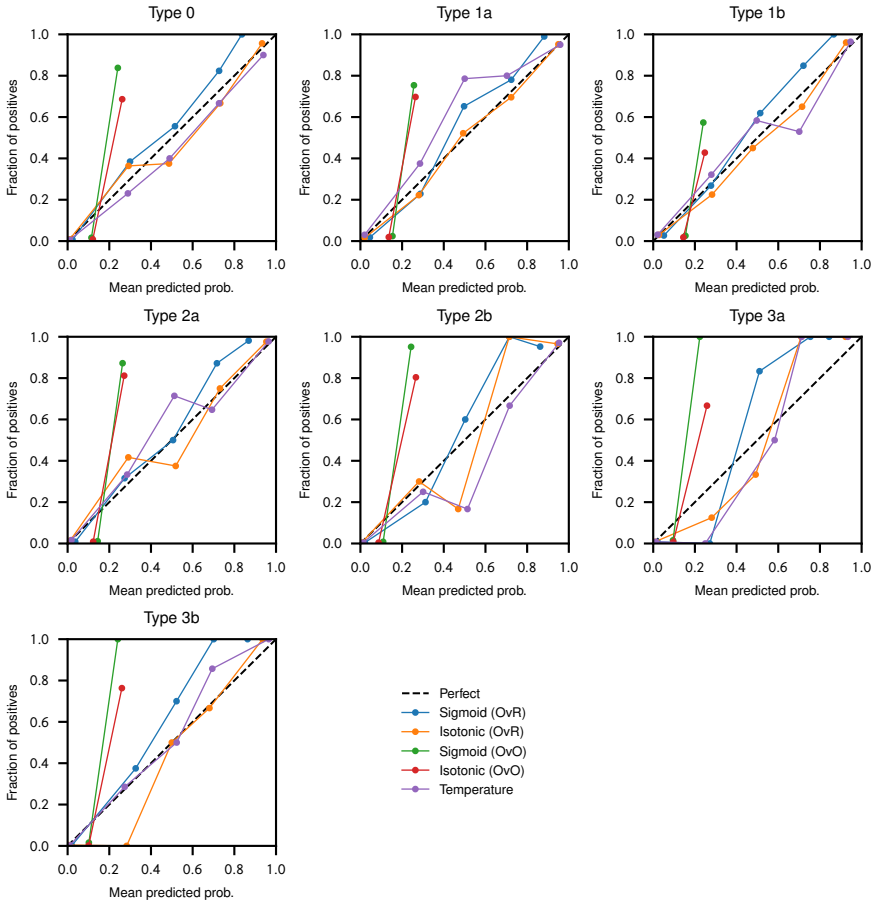


Figure 1: Calibration curves for each method, obtained by plotting the mean predicted probability against the true fraction of positive samples in each bin using a uniform binning strategy. The closer the curve is to the diagonal line, the better the calibration. Missing points indicate probability bins that contain no predicted samples.

the empirical fraction of positive samples in each bin. Bins that contain no predicted samples are omitted from the plot. OvR methods and temperature scaling produce curves that closely follow the diagonal, indicating good calibration.

To quantitatively evaluate the performance of each calibration method, we compute both classification metrics (F1 score, precision, recall, and ROC-AUC) and probabilistic calibration metrics (negative log-likelihood and Brier score). Let N denote the total number of test samples, K the number of classes, $\hat{L}_j = \arg\max_k \hat{p}_k(Y_j)$ the predicted label, L_j the true label, and $\hat{p}_k(Y_j)$ the calibrated probability that sample j belongs to class k. Classification metrics (F1 score, precision, recall, and ROC-AUC) assess how accurately the model assigns labels and are computed using weighted averaging, where the contribution of each class k is weighted by its sample count N_k. Probabilistic calibration metrics (NLL and Brier score) assess how well the predicted probabilities match the true label distribution. The following paragraphs provide the mathematical definitions of each metric.

Weighted F1 score. The F1 score is the harmonic mean of precision and recall [31]. The weighted F1 score averages the per-class F1 scores:

$$\mathrm{F1} = \sum_{k=1}^{K} \frac{N_k}{N} \cdot \frac{2\,\mathrm{Pr}_k \cdot \mathrm{Re}_k}{\mathrm{Pr}_k + \mathrm{Re}_k}, \tag{48}$$

where Pr_k and Re_k are the precision and recall for class k, defined below.

Weighted precision. Precision for class k measures the fraction of samples predicted as class k that truly belong to class k:

$$\mathrm{Pr}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k}, \tag{49}$$

where TP_k and FP_k are the numbers of true positives and false positives for class k, respectively. The weighted precision is $\mathrm{Precision} = \sum_{k=1}^{K} (N_k/N)\,\mathrm{Pr}_k$.

Weighted recall. Recall for class k measures the fraction of true class-k samples correctly identified:

$$\mathrm{Re}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k}, \tag{50}$$

where FN_k is the number of false negatives for class k. The weighted recall is $\mathrm{Recall} = \sum_{k=1}^{K} (N_k/N)\,\mathrm{Re}_k$.

Weighted ROC-AUC. The receiver operating characteristic area under the curve (ROC-AUC) [32] is computed using the one-versus-rest strategy. For each class k, the ROC curve is constructed by varying a decision threshold on $\hat{p}_k$, and the area under this curve (AUC_k) is computed. The weighted ROC-AUC is

$$\text{ROC-AUC} = \sum_{k=1}^{K} \frac{N_k}{N} \cdot \mathrm{AUC}_k. \tag{51}$$

Table 1: Performance of classification models with different calibration methods. Boldface values indicate the selected method, and underlined values indicate the best-performing method.

Method	F1	Precision	Recall	ROC-AUC	NLL	Brier
Sigmoid (OvR)	**0.88**	**0.89**	**0.88**	**0.98**	**0.44**	**0.03**
Isotonic (OvR)	0.87	0.88	0.87	0.98	0.63	0.03
Sigmoid (OvO)	0.86	0.87	0.86	0.98	1.39	0.09
Isotonic (OvO)	0.88	0.89	0.88	0.98	1.32	0.09
Temperature	0.88	0.89	0.88	0.98	0.38	0.03

Negative log-likelihood (NLL). The NLL, also known as log loss [33], evaluates the quality of the predicted probability distribution:

$$\mathrm{NLL} = -\frac{1}{N}\sum_{j=1}^{N} \log \hat{p}_{L_j}(Y_j). \tag{52}$$

NLL penalizes confident but incorrect predictions severely, making it particularly sensitive to the quality of probability calibration.

Brier score. The multiclass Brier score [34] measures the mean squared error between the predicted probabilities and the one-hot encoded true labels:

$$\mathrm{Brier} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{j=1}^{N}\big(\hat{p}_k(Y_j) - \mathbf{1}[L_j = k]\big)^2. \tag{53}$$

A lower Brier score indicates better-calibrated probability estimates.

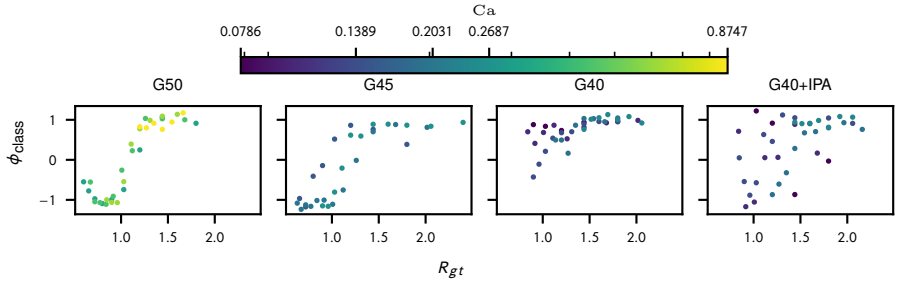
Table 1 shows the performance of each method, evaluated by stratified 5-fold cross-validation. Regarding classification accuracy, all methods achieve comparable performances, with sigmoid OvR achieving the highest scores for all four metrics. Regarding probabilistic calibration, temperature scaling achieves the lowest NLL (0.38) and Brier score (0.03), followed by sigmoid OvR (NLL 0.44, Brier 0.03). All OvO methods show significantly worse NLL and Brier scores compared to OvR methods and temperature scaling, indicating that the pairwise decomposition strategy leads to poorer probability estimates for our dataset. Hence, we select sigmoid OvR, isotonic OvR, and temperature scaling for further analysis in the following section.

3.3 Landscape quality

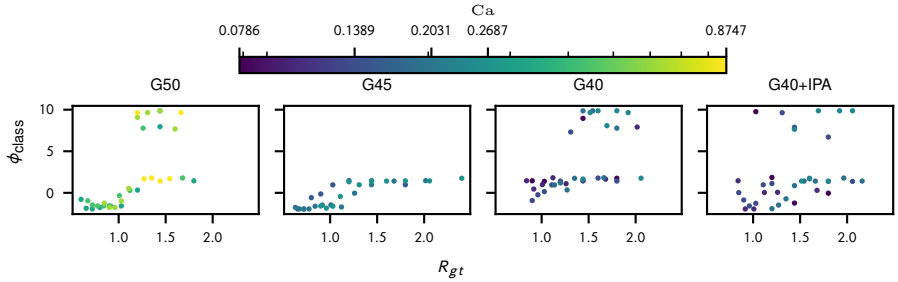
Figure 2 shows the ϕ_{class} values acquired from class probabilities calibrated by each method. Because sigmoid and temperature scaling fit probabilities using smooth parametric functions, the resulting ϕ_{class} landscapes are smooth and well-behaved. In contrast, isotonic regression fits a non-parametric step

Figure 2: Examples of ϕ_{class} values acquired from each calibration method.

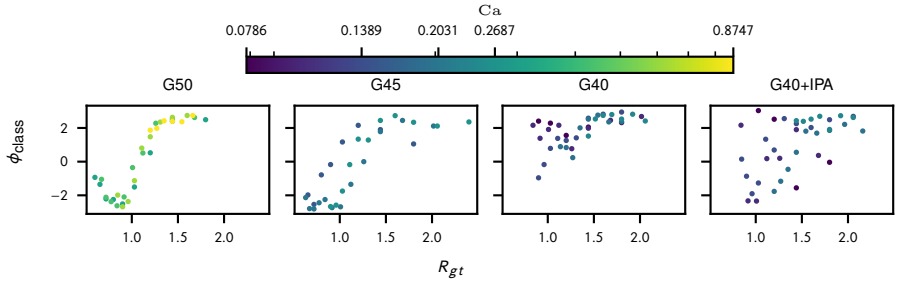
(a) Sigmoid OvR



(b) Isotonic OvR



(c) Temperature scaling



function, which leads to abrupt jumps in ϕ_{class} values for samples with similar scores, resulting in a more rugged landscape. This rugged landscape causes poor optimization and modeling performance in downstream tasks.

Figure 3 compares the benchmark results of Bayesian optimization using ϕ_{class} values from sigmoid OvR, isotonic OvR, and temperature scaling. It was already shown in the main text that α_{EI} outperforms other acquisition functions for sigmoid OvR. For isotonic OvR, α_{PI} shows the best performance, and for temperature scaling, α_{EI} again performs the best. When the three best-performing acquisition functions from each method are compared, isotonic OvR was significantly worse than the other two methods because of poor landscape quality. Sigmoid OvR slightly outperforms temperature scaling, hence we select sigmoid OvR as the final calibration method for our classification model.

4 Bayesian optimization

This section provides an overview of Bayesian optimization (BO). Bayesian optimization is characterized by a surrogate model and an acquisition function. We focus on BO with random forest (RF) surrogate model, as it is the method used in our study. Then, we discuss acquisition functions that guide the search for optimal solutions.

Let $\ell = \mathcal{L}_{\text{shape}}(Y) \in \mathbb{R}$ be the target value of optimization. Minimizing ℓ requires modeling the relationship

$$\ell = f(X) + \epsilon, \quad \epsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\text{noise}}), \tag{54}$$

where X is a d-dimensional input vector and σ^2_{noise} represents the noise variance. The optimization problem can then be formulated as finding

$$X_{\text{opt}} = \arg\min_X f(X). \tag{55}$$

When $f(X)$ is a black-box function, one can estimate a surrogate model from the collected data and use an acquisition function to iteratively guide the search for X_{opt}.

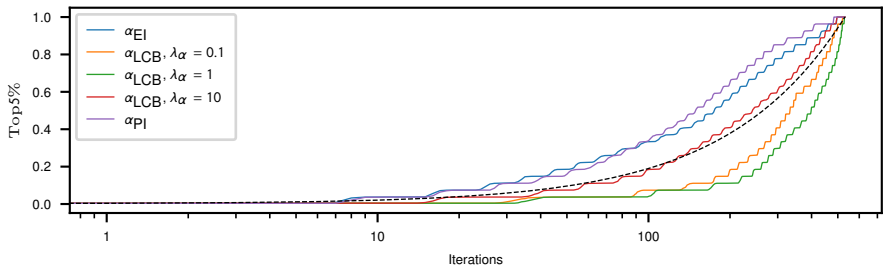
4.1 Random forest surrogate

Random forest is an ensemble of decision tree models, each trained on a bootstrap sample of the observed data $\mathcal{D}_n$. Random forest predictions are made by averaging the outputs of all trees in the ensemble. The advantage of RF-based BO is that it does not rely on explicit distributional assumptions about the latent function $f(X)$. Additionally, RF can naturally handle categorical inputs, which are common in materials design problems. Hence, we choose RF as surrogate model for BO in our study.

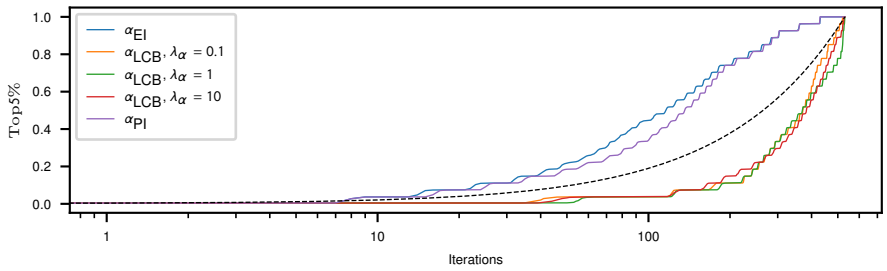
To enable Bayesian optimization with random forest, uncertainty quantification is essential for computing acquisition functions. The predictive uncertainty

Figure 3: Benchmark results of Bayesian optimization using ϕ_{class} values from each calibration method.

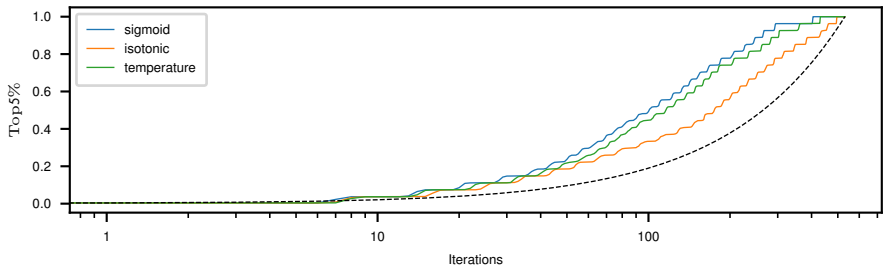
(a) Isotonic OvR



(b) Temperature scaling



(c) Comparison between best of each method



$\sigma_*^2(X_*)$ is estimated as the variance of predictions across individual trees in the ensemble:

$$\sigma_*^2(X_*) = \frac{1}{N_{\text{tree}} - 1} \sum_{i=1}^{N_{\text{tree}}} \left(f_i(X_*) - \mu_*(X_*)\right)^2, \tag{56}$$

where N_{tree} is the number of trees, $f_i(X_*)$ is the prediction from the i-th tree, and $\mu_*(X_*) = \frac{1}{N_{\text{tree}}} \sum_{i=1}^{N_{\text{tree}}} f_i(X_*)$ is the ensemble prediction. Note that RF-based BO is not fully Bayesian as no prior distribution is defined. Nevertheless, μ_* and σ_* can be used in acquisition functions as if they were statistics of a posterior distribution.

4.2 Acquisition function

Selection of the next evaluation point is guided by an acquisition function $\alpha(X)$ that balances exploration and exploitation. Given α, the next evaluation point is selected by:

$$X_{\text{new}} = \arg\max_X \alpha(X), \quad \ell_{\text{new}} = \mathcal{L}(Y_{\text{new}}). \tag{57}$$

The new point $(X_{\text{new}}, \ell_{\text{new}})$ is added to $\mathcal{D}_n$ to form $\mathcal{D}_{n+1} = (\mathbf{X}_{n+1}, \boldsymbol{\ell}_{n+1})$ with size $n+1$. This procedure is repeated for a fixed maximum number of iterations, and the optimal X yielding the smallest ℓ is selected.

A popular choice for $\alpha(X)$ is the expected improvement (EI) function [35]:

$$\alpha_{\text{EI}}(X) = E\left[\max(0, \ell_{n,\min} - f(X)) \mid \boldsymbol{\ell}_n\right], \tag{58}$$

where $\ell_{n,\min} = \min \boldsymbol{\ell}_n$. Assuming Gaussian posterior, EI can be computed in closed form as

$$\begin{aligned} \alpha_{\text{EI}}(X) = \; & (\ell_{n,\min} - \mu_*(X))\Phi\left(\frac{\ell_{n,\min} - \mu_*(X)}{\sigma_*(X)}\right) \\ & + \sigma_*(X)\varphi\left(\frac{\ell_{n,\min} - \mu_*(X)}{\sigma_*(X)}\right), \end{aligned} \tag{59}$$

where $\Phi(\cdot)$ and $\varphi(\cdot)$ are the cumulative distribution function and probability density function of the standard normal distribution, respectively. Other popular choices are the probability of improvement (PI) [36]

$$\alpha_{\text{PI}}(X) = \Phi\left(\frac{\ell_{n,\min} - \mu_*(X)}{\sigma_*(X)}\right), \tag{60}$$

and the lower confidence bound (LCB) [37]

$$\alpha_{\text{LCB}}(X) = \mu_*(X) - \lambda_\alpha \sigma_*(X), \tag{61}$$

where $\lambda_\alpha > 0$ is a hyperparameter that controls the exploration-exploitation ratio. High values of λ_α encourage exploration, and low values of λ_α encourage exploitation. PI is more exploitative, while EI adaptively balances exploration and exploitation based on the collected data.

5 Benchmarking of active learning algorithms

Let $\mathcal{D} = \{(X_j, \ell_j)\}_{j=1}^{M}$ be the pool of collected data points. In each simulation run, M_0 points are randomly drawn from $\mathcal{D}$ without replacement to form the initial dataset $\mathcal{D}_0$. The next point is then selected as $X_{\text{new}} = \arg\max_{X \in \mathcal{D} \setminus \mathcal{D}_0} \alpha(X)$, where α is the acquisition function. With the corresponding ℓ_{new}, the new point $(X_{\text{new}}, \ell_{\text{new}})$ is added to $\mathcal{D}_0$ to form $\mathcal{D}_1$. This procedure is repeated for $n = M - M_0$ iterations, resulting in $\mathcal{D}_n$, at which $\mathcal{D} \setminus \mathcal{D}_n = \emptyset$.

After $\mathcal{D}_n$ is constructed, the performance of the algorithm is evaluated. We adopt three performance metrics [38]:

$$\text{Top5\%}(j) = \frac{|\mathcal{D}_j \cap \mathcal{D}^{5\%}|}{|\mathcal{D}^{5\%}|}, \tag{62}$$

$$\text{EF}(j) = \frac{\text{Top5\%}_{\text{BO}}(j)}{\text{Top5\%}_{\text{RS}}(j)}, \tag{63}$$

$$\text{AF}(a) = \frac{\text{Top5\%}_{\text{RS}}^{-1}(a)}{\text{Top5\%}_{\text{BO}}^{-1}(a)}, \tag{64}$$

where $j = 1, \ldots, n$ and $\mathcal{D}^{5\%} \subset \mathcal{D}$ is the set of the top 5% points with the smallest ℓ. The enhancement factor $\text{EF}(j)$ measures how many more top 5% points are found by BO compared to random sampling (RS) after j iterations. The acceleration factor $\text{AF}(a)$ measures how many fewer iterations BO needs to find a given fraction of $\mathcal{D}^{5\%}$ compared to RS. The last two metrics require a statistical baseline $\text{Top5\%}_{\text{RS}}(j)$, which can be analytically computed [38]. The simulation is repeated S times with a different $\mathcal{D}_0$ each time, and the resulting metrics are bootstrapped with B resamples to estimate their medians and 95% confidence intervals. In this study, we set $S = 1,000$ and $B = 10,000$.

For all S simulations, we correct the $\text{Top5\%}_{\text{BO}}(j)$ value as follows:

$$\text{Top5\%}_{\text{BO}}(j) = \begin{cases} \text{Top5\%}_{\text{RS}}(j) & \text{if } j \leq M_0, \\ \max\left(\dfrac{|\mathcal{D}_j \cap \mathcal{D}^{5\%}|}{|\mathcal{D}^{5\%}|}, \text{Top5\%}_{\text{RS}}(M_0)\right) & j > M_0. \end{cases} \tag{65}$$

This is because the first M_0 iterations of Bayesian optimization are driven by the initial random sampling (RS). After correction, the sampling distributions of $\text{Top5\%}_{\text{BO}}$, EF, and AF are obtained.

6 Edge formation mechanism

6.1 Effects of IPA co-solvent

In our dataset, the G40 and G40+IPA slurries differ only by the addition of 2 wt % IPA to the latter. The most readily measurable consequence of this composition change is a reduction of the static surface tension from $\sigma = 75\,\text{mN/m}$ to $\sigma = 57\,\text{mN/m}$ (see tables and figures in the Methods section of the main

text). This reduction in σ is consistent with increased wettability of the slurry on the substrate, which decreases from $\theta = 72.8°$ to $\theta = 62.4°$. Other properties, such as density and shear viscosity, are nearly identical between the two slurries. However, the two slurries show a noticeable difference in edge profile quality, signifying that the slurry–substrate interaction plays an important role in edge formation.

There are other possible mechanisms by which the IPA addition could affect edge formation. For example, the surface tension gradients $\partial\sigma/\partial\xi$ could in principle drive Marangoni flows near the edge, but these have been reported to be negligible for the present dataset [8]. The altered interfacial affinity can also modify the wetting behavior on slot-die lips, which can affect the initial shape of the coating layer. This effect is difficult to decouple from current dataset, and direct confirmation would require additional measurements such as wetting dynamics imaging. However, difference of shape metrics between slurries are well-described by the contribution of $\cos\theta$ in current analysis, and unexplained residuals are captured by Gaussian process models.

6.2 The Schmitt–Dobroth model

In the main text, we employ the Schmitt–Dobroth model [5] to construct a physics-informed prior mean for nonparametric model of H_{app}. The interpretation below combines Schmitt's original model with our own theoretical synthesis of its limiting behavior. We emphasize that this formulation is a semi-empirical, reduced-order model rather than a first-principles derivation from fundamental fluid mechanics or contact-line physics.

Dobroth and Erwin's free-film stretching argument implies

$$\frac{h_{\max}}{H_{\mathrm{g}}} \sim \frac{1}{\sqrt{R_{\mathrm{gt}}}}, \tag{66}$$

which gives the baseline trend that edge elevation increases with the stretching intensity R_{gt}. If $h_0 \approx h_{\mathrm{w}} = H_{\mathrm{g}}/R_{\mathrm{gt}}$, the same argument yields $H_{\mathrm{app}} \sim \sqrt{R_{\mathrm{gt}}}$. Schmitt's contribution is to add an empirical parameter λ to account for the effect of substrate by,

$$x^2 + \lambda x - \frac{1}{R_{\mathrm{gt}}} = 0, \qquad x := \frac{h_{\max}}{H_{\mathrm{g}}}, \tag{67}$$

which yields the closed-form relation

$$\frac{H_{\mathrm{g}}}{h_{\max}} = \frac{2}{-\lambda + \sqrt{\lambda^2 + 4/R_{\mathrm{gt}}}} \tag{68}$$

used in the main text. In this form, the x^2 term recovers the Dobroth free-film limit and the λx term acts as a correction for substrate interaction effects.

Schmitt originally referred to λ as an *adhesion parameter*. In the present manuscript, however, we use the broader term *substrate interaction parameter*. This change is to stress that λ lumps several different effects, including

substrate wettability, contact-line pinning, coating-substrate interfacial affinity, capillarity-mediated reshaping near the contact line, and the resulting resistance to free-edge lateral contraction. This interpretation is also supported by the behavior of the model itself. The positive solution of the quadratic relation is

$$x(\lambda) = \frac{-\lambda + \sqrt{\lambda^2 + 4/R_{\mathrm{gt}}}}{2}, \tag{69}$$

and its derivative satisfies

$$\frac{\mathrm{d}x}{\mathrm{d}\lambda} = \frac{-1 + \lambda/\sqrt{\lambda^2 + 4/R_{\mathrm{gt}}}}{2} < 0. \tag{70}$$

Hence, increasing λ monotonically suppresses the predicted edge height. Two limiting regimes are particularly useful. For $\lambda^2 \gg 4/R_{\mathrm{gt}}$,

$$\frac{h_{\max}}{H_{\mathrm{g}}} \approx \frac{1}{\lambda R_{\mathrm{gt}}}, \qquad H_{\mathrm{app}} \approx \frac{1}{\lambda}, \tag{71}$$

so the edge height is controlled more directly by substrate interaction than by stretching. For weak substrate interaction,

$$H_{\mathrm{app}} \approx \sqrt{R_{\mathrm{gt}}} - \frac{\lambda R_{\mathrm{gt}}}{2}, \tag{72}$$

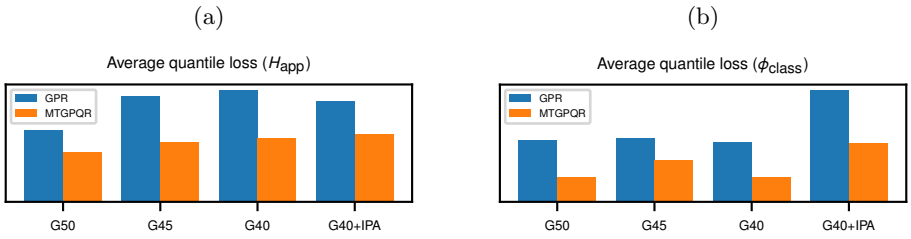
showing that even modest changes in λ can noticeably suppress edge elevation at large R_{gt}.

Schmitt placed a simple linear model $\lambda = aR_{\mathrm{gt}} + b$ and empirically fitted a and b. However, in this study we found little correlation between λ and R_{gt}, and instead discovered that λ nonlinearly depends on Ca and $\cos\theta$. To agree with this observation, we model λ by $\lambda(\mathrm{Ca}, \cos\theta) = a\mathrm{Ca}^b(\cos\theta)^c$ and empirically fit a, b, and c. Although not being a rigid mechanistic model, this semi-empirical formulation provides a better fit to the data than Schmitt's original linear model, and it captures the observed trends in the data. But mismatching residuals and uncertainties can be learned by the Gaussian process approach, allowing the model to be a suitable physics-informed inductive bias.

7 Ablation examples of MTGPQR

The multi-task Gaussian process quantile regression (MTGPQR) model is designed to leverage both the physical prior of Gaussian process regression and uncertainty estimation of non-crossing quantile regression. Three design choices are important: (i) the additive structure with softplus-enforced gap functions ΔQ that guarantee non-crossing quantiles, (ii) the linear model of coregionalization (LMC) kernel that captures correlations between quantile levels, and (iii) the physics-based prior mean that guides extrapolation in data-sparse regions. This section demonstrates the importance of each component by comparing the full MTGPQR model against ablated variants in which one component is removed at a time.

Figure 4: Quantile estimation performances of the standard Gaussian process regression (GPR) and the multi-task Gaussian process quantile regression (MTGPQR). Multiple quantiles with different levels are estimated for H_{app} and ϕ_{class}, and average values of quantile losses from training data for each model are plotted. Both models for each shape metric are equipped with the same prior mean, kernel structure and lengthscale constraints.



7.1 Comparison to the standard GPR

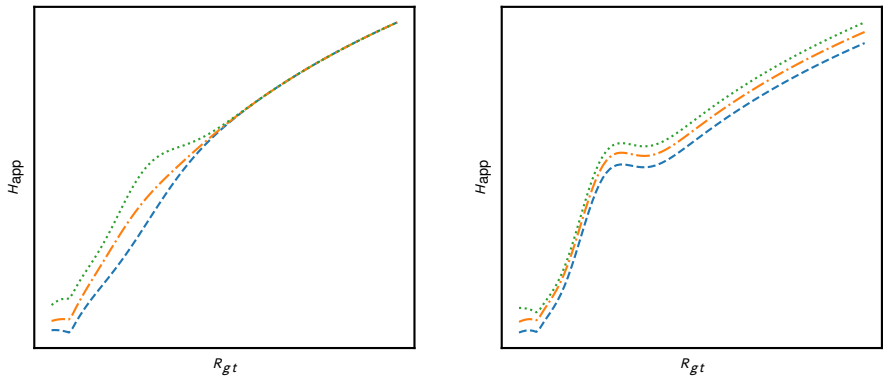
We first examine whether quantile regression is truly necessary. As described in Supplementary Section 2.2, the standard Gaussian process regression (GPR) model provides the Gaussian posterior distribution, from which arbitrary quantiles can be computed. However, this approach assumes that the residuals of the mean function are homoscedastic and Gaussian, which may not hold in practice.

Figure 4 compares the quantile estimation performance of the GPR and MTGPQR models for the two shape metrics H_{app} and ϕ_{class}. The average quantile loss, which measures the calibration of the estimated quantiles against the training data, is plotted for each model and shape metric. It can be observed that MTGPQR consistently achieves lower quantile losses than GPR, indicating that it provides better-calibrated quantile estimates. This improvement can be attributed to the fact that MTGPQR directly models the quantiles with fewer assumptions, making it more flexible in capturing the probabilistic structure of the data. In the following ablation studies, we focus on the MTGPQR model for H_{app}.

7.2 Removing additive structure

When the additive structure of the MTGPQR model is removed and each quantile is individually estimated, the result can suffer from the quantile crossing phenomenon. Even if quantiles are non-crossing, they can collapse to a degenerate zero-width distribution in extrapolation regions. Figure 5 compares an ordinary Gaussian process quantile regression (GPQR) model in which each quantile is estimated by a separate, independent GP, to the MTGPQR. Although GPQR does not exhibit quantile crossing, its quantiles collapse to the same value in the extrapolation region where R_{gt} is high. As a result, no probabilistic information can be extracted from the quantiles in this region. On the other hand, the MTGPQR model maintains a constant gap between quantiles in the extrapola-

Figure 5: Comparison between GPQR and MTGPQR models for H_{app}, using 0.05, 0.5, and 0.95 quantiles. All Gaussian processes of GPQR are equipped with the physical prior mean of H_{app}, causing the quantiles to collapse to identical values in the large-R_{gt} region. MTGPQR, on the other hand, models the gaps between quantiles with constant priors, allowing the quantiles to maintain a constant width in the same region.



(a) GPQR (b) MTGPQR

tion region due to the additive structure of the gap functions. As a result, the MTGPQR model provides meaningful uncertainty estimates even in data-sparse regions.

7.3 Removing intertask correlation

MTGPQR combines latent GPs to construct gap functions. There are two ways to relate the latent GPs to the gap functions. The linear model of coregionalization (LMC) models the gap functions as linear combinations of shared latent GPs. On the other hand, an independent multitask model assigns a separate latent GP to each gap function with no shared structure.

Figure 6 compares the quantile estimation performance of the LMC and independent multitask models for the two shape metrics H_{app} and ϕ_{class}. It can be observed that, for most cases, LMC achieves lower losses than the independent multitask model. This result can be explained by the fact that LMC learns the correlation structure between gaps, providing more flexibility in modeling the quantiles. In fact, the independent multitask model can be seen as a special case of LMC where the coregionalization matrix is diagonal, so it is less expressive than LMC.

Figure 6: MTGPQR with different intertask correlation structures. Multiple quantiles with different levels are estimated for H_{app} and ϕ_{class}, and average values of quantile losses from training data for each model are plotted. Both models for each shape metric are equipped with the same prior mean, kernel structure, lengthscale constraints, number of quantiles and latent GPs.

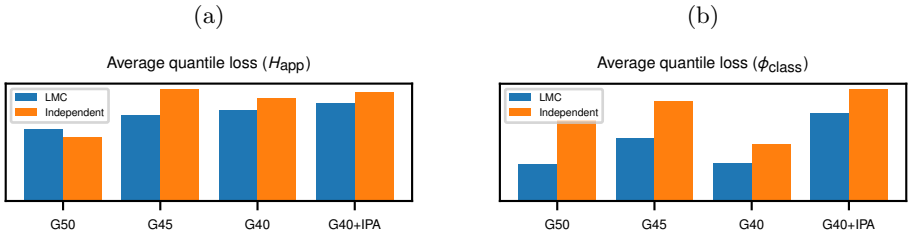
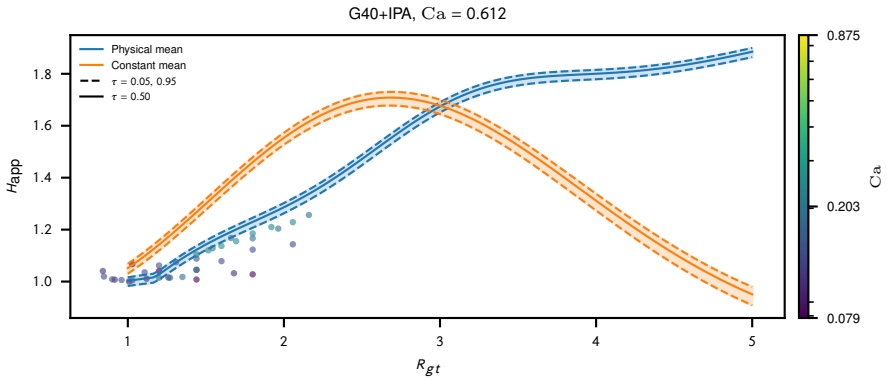


Figure 7: Comparison of the physical prior mean and constant prior mean in MTGPQR.



7.4 Removing physical prior

Figure 7 compares the MTGPQR model with the physics-based prior mean against a variant that uses a constant prior mean. With the physical prior mean, quantiles follow the monotonically increasing behavior of the shape metric, even in extrapolation regions where no training data are available. With a constant prior mean, however, the quantiles eventually converge to a constant value in the extrapolation region, which is inconsistent with the known physical behavior of the system. These results demonstrate the importance of incorporating physical priors to ensure physically consistent predictions.

References

[1] T Dobroth and Lewis Erwin. Causes of edge beads in cast films. *Polymer Engineering & Science*, 26(7):462–467, 1986.

[2] Edgar B Gutoff and Edward D Cohen. *Coating and drying defects: troubleshooting operating problems*. John Wiley & Sons, 2006.

[3] Marcel Schmitt, Philip Scharfer, and Wilhelm Schabel. Slot die coating of lithium-ion battery electrodes: investigations on edge effect issues for stripe and pattern coatings. *Journal of Coatings Technology and Research*, 11(1):57–63, 2014.

[4] Boris Bitsch, Jens Dittmann, Marcel Schmitt, Philip Scharfer, Wilhelm Schabel, and Norbert Willenbacher. A novel slurry concept for the fabrication of lithium-ion battery electrodes with beneficial properties. *Journal of Power Sources*, 265:81–90, 2014.

[5] Marcel Schmitt. *Slot die coating of lithium-ion battery electrodes*. KIT Scientific Publishing, 2016.

[6] Sandro Spiegel, Thilo Heckmann, Andreas Altvater, Ralf Diehm, Philip Scharfer, and Wilhelm Schabel. Investigation of edge formation during the coating process of li-ion battery electrodes. *Journal of Coatings Technology and Research*, 19(1):121–130, 2022.

[7] Sandro Spiegel, Alexander Hoffmann, Julian Klemens, Philip Scharfer, and Wilhelm Schabel. Optimization of edge quality in the slot-die coating process of high-capacity lithium-ion battery electrodes. *Energy Technology*, 11(5):2200684, 2023.

[8] Jihwan Yoon, Kyengmin Min, Jisoo Song, and Jaewook Nam. Controlling slot-coating edge profiles of battery electrodes. *ACS Applied Materials & Interfaces*, 2025.

[9] Brian G Higgins and Lawrence E Scriven. Capillary pressure and viscous pressure drop set bounds on coating bead operability. *Chemical Engineering Science*, 35(3):673–682, 1980.

[10] Alexis Boukouvalas, Remi Barillec, and Dan Cornford. Gaussian process quantile regression using expectation propagation. *arXiv preprint arXiv:1206.6391*, 2012.

[11] Matthew A Taddy and Athanasios Kottas. A bayesian nonparametric approach to inference for quantile regression. *Journal of Business & Economic Statistics*, 28(3):357–369, 2010.

[12] Nicolai Meinshausen and Greg Ridgeway. Quantile regression forests. *Journal of machine learning research*, 7(6), 2006.

[13] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

[14] Yandong Yang, Shufang Li, Wenqi Li, and Meijun Qu. Power load probability density forecasting using gaussian process quantile regression. *Applied Energy*, 213:499–509, 2018.

[15] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[16] Keming Yu and Rana A Moyeed. Bayesian quantile regression. *Statistics & Probability Letters*, 54(4):437–447, 2001.

[17] Ichiro Takeuchi, Quoc Le, Timothy Sears, and Alexander Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7(45):1231–1264, 2006.

[18] Kristian Lum and Alan E Gelfand. Spatial quantile multiple regression using the asymmetric laplace process. *Bayesian Analysis*, 7(2):235–258, 2012.

[19] Songfeng Zheng. Qboost: Predicting quantiles with boosting for regression and binary classification. *Expert Systems with Applications*, 39(2):1687–1697, 2012.

[20] James W Taylor. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of forecasting*, 19(4):299–311, 2000.

[21] Sanket R Jantre, Shrijita Bhattacharya, and Tapabrata Maiti. Quantile regression neural networks: a bayesian approach. *Journal of Statistical Theory and Practice*, 15(3):68, 2021.

[22] Hideo Kozumi and Genya Kobayashi. Gibbs sampling methods for bayesian quantile regression. *Journal of statistical computation and simulation*, 81(11):1565–1578, 2011.

[23] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[24] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[25] Sachinthaka Abeywardana and Fabio Ramos. Variational inference for nonparametric bayesian quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[26] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

[27] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[28] Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5(Aug):975–1005, 2004.

[29] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[30] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[31] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[32] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[33] Irving John Good. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, 1952.

[34] W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[35] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[36] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 03 1964.

[37] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[38] Qiaohao Liang, Aldair E Gongora, Zekun Ren, Armi Tiihonen, Zhe Liu, Shijing Sun, James R Deneault, Daniil Bash, Flore Mekki-Berrada, Saif A Khan, et al. Benchmarking the performance of bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, 2021.