Results in Engineering 29 (2026) 109409

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

Enhancing design of experiments through uncertainty estimation and synthetic data generation

Luis Moles [a,b,*], Alain Andres [a,c], Goretti Echegaray [b], Fernando Boto [c]

[a] TECNALIA, Basque Research and Technology Alliance (BRTA), Donostia, 20009, Spain
[b] University of the Basque Country (UPV/EHU), Donostia, 20018, Spain
[c] University of Deusto, Donostia, 20012, Spain

ARTICLE INFO

Keywords:
Data augmentation
Design of experiments
Gaussian process
Synthetic data
Uncertainty estimation

ABSTRACT

Design of Experiments is a key methodology for optimizing machine learning models, but traditional methods often depend on extensive real data collection, which is costly and time-consuming. Moreover, predefined experimental designs may struggle at adapting to complex or high-dimensional input spaces, sometimes leading to inefficient exploration, especially when data are scarce and uncertainty is high. To address these challenges, we propose a methodology that integrates uncertainty estimation with synthetic data generation. First, we evaluate several uncertainty estimators (Gaussian Process, Monte Carlo Dropout and Tree-based ensembles) which identify the input regions where the current model is most uncertain. Next, we analyze different generative models (Variational Autoencoders, Generative Adversarial Networks, and Large Language Models) trained under varying levels of data availability (from only 10% of the real dataset up to full data), to test their robustness in extreme scarcity conditions. Finally, we combine the best uncertainty estimator with the most reliable generative model in a hybrid active learning pipeline. Beyond the standard setting, we systematically vary the number and proportion of synthetic versus real samples, showing how the mixture affects predictive accuracy and uncertainty reduction. Results of the experimentation show that Gaussian Process uncertainty estimation outperforms other tested methods under extreme data scarcity, and that Variational Autoencoders produce the most stable synthetic samples with as little as 10% of the real data used for training. The full hybrid loop (Gaussian Process + Variational Autoencoder) achieves similar R^2 to baselines while driving down uncertainty significantly faster, offering a data-efficient strategy for costly experimental contexts.

1. Introduction

Design of Experiments (DoE) is a fundamental methodology for systematically exploring input variables to gain insights into system behavior and optimize performance across various engineering and industrial domains [1]. In the context of machine learning prediction tasks where data is collected from a specific domain, DoE plays a crucial role in constructing informative datasets that support accurate and robust model training. By carefully selecting experimental conditions, researchers aim to maximize the predictive power of their models while minimizing the number of required experiments. However, in many real-world scenarios data collection is often costly, time-consuming, and resource-intensive, making traditional DoE approaches difficult to implement.

One of the primary limitations of traditional DoE techniques, such as factorial designs [2], response surface methods [3], and Latin hypercube sampling [4], is that their reliance on a predefined sampling strategy does not dynamically adapt to the complexity of the underlying model or to changes in the data distribution over time [5]. In data-scarce scenarios, where the number of real experiments is constrained by cost, time, or feasibility, such rigid sampling approaches may lead to suboptimal model performance due to an inadequate exploration of the most informative regions of the input space. This may result in poor generalization, higher prediction uncertainty, and an incomplete understanding of the system under study.

To address these challenges, active learning frameworks that dynamically select new experiments based on the model's uncertainty have been studied [6,7]. Such approaches continually evaluate model confidence to decide which data points to acquire next, and has been applied effectively in different domains [8,9]. In order to quantify uncertainty at each point, techniques such as Gaussian Processes (GP) [10] and Monte Carlo Dropout (MC-Dropout) [11] have been extensively employed. These methods provide insights into the model's confidence in its predictions across different regions of the input space. In the context of DoE for machine learning, uncertainty estimation helps identify

* Corresponding author.
E-mail address: luis.moles@tecnalia.com (L. Moles).

https://doi.org/10.1016/j.rineng.2026.109409
Received 19 September 2025; Received in revised form 28 January 2026; Accepted 1 February 2026
Available online 5 February 2026

areas where the model lacks sufficient knowledge, often due to limited or biased data coverage. By quantifying uncertainty, we can prioritize data collection in the most informative areas, ensuring that additional experiments contribute meaningfully to improving model performance rather than redundantly sampling well-explored areas.

However, as previously mentioned, collecting real data in these uncertain regions is often costly or impractical. In such cases, data augmentation techniques provide a powerful alternative by generating synthetic data to expand the available dataset without the need for additional real-world experiments. Several well-established methods for synthetic data generation have been developed to address this issue. Classical approaches like SMOTE [12] and its variants have long been used to handle data scarcity by interpolating between existing samples. More recently, deep learning-based methods for synthetic data generation have emerged as powerful tools. Unlike many pipelines focused on images, our work addresses the tabular data domain, which presents unique challenges in maintaining realistic feature correlations. In particular, generative modeling techniques such as Generative Adversarial Networks (GANs) [13] and Variational Autoencoders (VAEs) [14] learn to approximate the real data distribution and generate new synthetic samples. Even Large Language Models (LLMs) [15] emerged as a viable option for generating quality artificial data. However, these strategies rely on complex and deep architectures, which makes their effectiveness particularly challenging in data-scarce scenarios, where there is already a limited amount of real data to generate reliable synthetic samples. Overcoming this limitation may require a more targeted approach, where the synthetic data is generated in a principled manner (guided by uncertainty estimation) to focus on the most informative and underexplored regions of the input spaces of the data space.

Therefore, in this paper we propose a methodology for DoE that accelerates the reduction of the model's predictive uncertainty by strategically integrating uncertainty estimation and synthetic data generation. Rather than relying on predefined experimental sampling strategies, our approach dynamically identifies the regions of highest uncertainty in the input space, and generates high-quality synthetic data specifically in those zones to enrich model learning. By leveraging uncertainty-guided synthetic data augmentation, we ensure that the most informative areas are addressed, leading to more targeted and efficient experimental designs. The experiments are based on a real-world dataset from a femtosecond laser texturing process, where each simulation requires extensive computational resources and time. As a result, only a limited number of data points is available. This data scarcity highlights the need for a methodology that can efficiently focus sampling efforts and augment the dataset through synthetic samples, targeting areas of high uncertainty. Building on this scenario, our goal is to contribute to the field in three different ways, which can be summarized as follows:

- Contribution 1 (C1): We evaluate how different uncertainty estimation techniques (GP, MC-Dropout and Tree-based Ensembles) can be leveraged to identify the most informative regions of the input space, optimizing the selection of experimental data points via active learning.
- Contribution 2 (C2): We investigate the effectiveness of synthetic data augmentation using three different deep learning architectures (GAN, VAE, LLM), assessing their ability to generate high-quality synthetic samples that enhance model performance, especially when real data is limited.
- Contribution 3 (C3): We introduce a hybrid methodology that integrates uncertainty estimation and synthetic data generation within an active learning loop, by dynamically selecting from a family of generative models pretrained on increasing fractions of the real dataset.
- Contribution 4 (C4): We analyze extreme scenarios (synthetic only updates) and varying synthetic adding ratios, showing that data augmentation can significantly reduce the uncertainty.

The structure of the paper is organized as follows. Section 2 reviews the state of the art, focusing on uncertainty estimation, synthetic data generation, and DoE in data-scarce scenarios. Section 3 details the materials and methods, including the uncertainty estimation techniques, the synthetic data generation architectures, and the overall proposed methodology, while Section 4 describes the experimental setup, covering dataset, hyperparameters, baselines and evaluation metrics. Section 5 presents and discusses the experimental results. Finally, Section 6 provides the conclusions, summarizing key findings and proposing directions for future research. Appendices report extended results and additional visualizations.

2. Related work

This section provides a comprehensive overview of the research related to our work, structured into three main parts. Section 2.1 discusses the foundations of DoE in the context of machine learning, with a particular emphasis on the role of uncertainty estimation as a tool to guide data acquisition in data-scarce scenarios. Section 2.2 focuses on synthetic data generation and augmentation techniques reviewing commonly adopted approaches, such as generative models, and discussing their applicability and limitations in low-data regimes and Section 2.3 summarizes our contribution within the context of related work.

2.1. Design of experiments and uncertainty estimation

The integration of DoE with machine learning methodologies has led to new strategies for optimizing data collection in predictive modeling tasks. Traditionally, DoE approaches such as factorial design, response surface methods, and latin hypercube sampling have structured the sampling process. However, these approaches often assume static, uniform strategies and may struggle when applied to high-dimensional, complex problems encountered in modern machine learning applications.

A growing body of work investigates how machine learning can complement and even enhance traditional DoE. In particular, the study by Freiesleben et al. [16] explores whether ML might replace or reinforce DoE. The authors conclude that ML and DoE are complementary approaches, and they highlight the potential of combining the two methodologies to improve process optimization and quality management. This reflects a broader shift toward hybrid approaches, suggesting a redefinition of DoE as a dynamic, model-informed methodology instead of a static statistical tool.

One of the key enablers of this shift is the use of Bayesian Optimization (BO). Greenhill et al. [17] review the application of BO in DoE tasks. They show how BO incorporates prior knowledge, handles multiple objectives, and adapts to complex experimental spaces, including high-dimensional, constrained, and multi-fidelity scenarios. BO offers a probabilistic framework that can leverage model uncertainty to identify the most informative points for data collection, ultimately leading to more efficient and targeted experimental designs.

A notable contribution in this direction is the incremental DoE methodology proposed in Voigt et al. [5]. This work presents a stepwise experimental design framework that combines GP Regression with expert knowledge. Their approach incrementally extends the experimental space by prioritizing inputs and designing new experiments step by step, achieving high model quality even with limited data. While not based on uncertainty-driven acquisition, this iterative, accuracy-driven refinement aligns with our methodology, which adaptively guides synthetic data selection based on model feedback.

Similarly, Malluhi et al. [18] propose a workflow integrating BO with variance-based acquisition functions and stopping criteria based on uncertainty convergence. They introduce empirical criteria to determine when further experiments yield diminishing returns, using prediction uncertainty as a stopping signal. This study reinforces the role of uncertainty estimation as a central component in adaptive, resource-efficient DoE strategies.

More recently, researchers have explored the broader role of uncertainty-guided learning beyond real data acquisition. For instance, Ji et al. [19] propose a method that uses uncertainty maps derived from both model and data uncertainty to improve training efficiency and reliability in a deep learning task. Although not focused specifically on DoE, this work illustrates how uncertainty estimation can guide model refinement and data handling together, supporting our approach of using uncertainty not only to select real samples but also to guide synthetic data generation and validation.

2.2. Synthetic data generation

The growing complexity of deep learning models increasingly demands large and diverse datasets, posing a significant challenge in data-scarce or costly acquisition scenarios. In response, synthetic data generation and data augmentation techniques have attracted significant attention as strategies to expand training datasets and improve model performance without increasing experimental effort. A key issue explored in the literature is the effectiveness of generative models in low-data regimes, where training instability and overfitting may arise. Generative modeling paradigms such as GANs, VAEs, normalizing flows, diffusion models, and more recently LLM-based generators, have demonstrated effectiveness across a wide range of disciplines, including computer vision, natural language processing, scientific simulation, and tabular data modeling.

GAN-based methods have been actively investigated under limited data availability. Karras et al. [20] propose an adaptive discriminator augmentation strategy that stabilizes GAN training with only a few thousand samples, while Gurumurthy et al. [21] introduce the DeLiGAN model, which improves sample diversity via mixture-model reparameterization of the latent space. These methods demonstrate the viability of GANs in data-scarce contexts, supporting their integration into our proposed methodology.

Beyond image data, recent research has increasingly addressed synthetic data generation for tabular datasets, which are more prevalent in engineering and scientific applications. Apellániz et al. [22] incorporate inductive bias through transfer and meta-learning to improve generative performance from small tabular datasets, and Delgado and Oyedele [23] show that multiple autoencoder variants can generate synthetic data that improves downstream predictive performance. These studies support the use of GAN and VAE based generators as practical tools for structured data augmentation under limited data availability.

The synergy between synthetic data generation and active learning has also been explored. Moles et al. [24] review this interaction in supervised learning, highlighting that synthetic data can improve generalization in underrepresented regions of the input space, while Mosqueira-Rey et al. [25] combine synthetic generation with expert-guided active learning in a medical context. Although these works focus on tasks that differ from experimental design, they motivate iterative uncertainty-guided augmentation strategies.

In addition to GANs and VAEs, other probabilistic generative modeling paradigms have been proposed and reviewed in the literature. Normalizing flows [26] learn flexible density models through sequences of invertible transformations and enable exact likelihood estimation, while diffusion-based models [27] learn a denoising process that can achieve high-fidelity generation. Comprehensive reviews such as that of Papamakarios et al. [28] and Yang et al. [29] highlight the broad applicability of these approaches across multiple domains.

Recent advancements in LLMs have positioned them as a viable alternative to traditional generative models for synthetic data generation. Studies such as [30] and [31] explore the use of LLMs for generating high-quality tabular data, highlighting their ability to closely match real data distributions and, in some cases, outperform GANs in predictive tasks. Building on these findings, the recent GReaT framework [32] further demonstrates the promise of transformer-based generators for tabular synthesis. However, these approaches also introduce new challenges such as higher computational requirements. In our work, we aim to assess whether LLMs can be effectively leveraged in data-scarce scenarios for uncertainty-guided synthetic data generation and how they compare to more established techniques, such as CTGANs and VAEs.

In summary, while a variety of techniques have been proposed for generating synthetic data in low-data environments, most existing studies focus on either improving generative model architectures or applying synthetic data to enhance model training. Few explicitly integrate synthetic data generation into experimental design frameworks guided by model uncertainty. Our methodology fills this gap by combining uncertainty-driven sampling with multi-architecture synthetic data generation, offering a robust solution to the challenges of data scarcity in experimental design contexts.

2.3. Our contribution

While previous research has explored uncertainty-driven experimental design and synthetic data augmentation independently, our work addresses several gaps in the existing literature. First, we provide a systematic analysis of how uncertainty estimation can effectively guide experimental design decisions, particularly in data-scarce environments where traditional DoE methods may be insufficient. Second, we investigate the generation and validation of synthetic data using various deep learning architectures, offering insights into their relative performance and limitations in low-data scenarios. Finally, we propose a hybrid DoE methodology that incorporates uncertainty estimation and synthetic data generation to explore under-represented regions in extreme data scarce scenarios. Unlike previous studies, our approach treats data generation and selection not as isolated processes, but rather as interconnected steps in an adaptive, model-informed experimental design loop.

3. Materials and methods

In this section, we describe the methodological components that support our approach. Section 3.1 presents the uncertainty estimation techniques used to identify high-uncertainty regions in the input space. Section 3.2 introduces the synthetic data generation techniques evaluated in this work and Section 3.3 outlines the proposed hybrid methodology, showing how uncertainty estimation and data augmentation are integrated into an active learning loop.

3.1. Uncertainty estimation techniques

A central component of our methodology is the use of uncertainty estimation to guide the DoE and identify the most informative regions in the input space. In this work, we compare three distinct strategies to quantify total uncertainty. Each of these techniques provides a per-sample measure of predictive uncertainty[1], which is then used to guide the selection of new data points in our active-learning loop (see Section 3.3).

Gaussian Process Regression. GP regression [33] treats the mapping from inputs to outputs as a draw from a multivariate normal process rather than a fixed, deterministic function. At its core is a *covariance kernel* $k(x, x')$, which encodes our belief about how similar any two inputs x and x' should be (points closer in the input space are expected to have more strongly correlated outputs).

Formally, GP regression assumes that for any finite collection of inputs $\{x_i\}_{i=1}^n$, the corresponding outputs $\{f(x_i)\}_{i=1}^n$ follow a joint Gaussian distribution. Given observed data $D = \{(x_i, y_i)\}_{i=1}^n$, the GP posterior

[1] Uncertainty estimates are interpreted as predictive uncertainty, which may include both epistemic and aleatory contributions. We do not attempt to explicitly decompose uncertainty into its constituent sources; instead, all methods are compared consistently under the same data and training conditions.

provides, for each new input x_*, a predictive mean $\mu(x_*)$ (the best estimate of $f(x_*)$) and a predictive variance $\sigma^2(x_*)$ (which reflects predictive uncertainty due to limited data).

In this work, the GP posterior predictive variance $\sigma^2(x_*)$ is directly used as the uncertainty measure to guide sample selection, with larger values indicating regions of the input space that are less supported by the available data and therefore prioritized during the acquisition process.

A common choice for the kernel is the radial basis function (RBF):

$$k(x, x') = \sigma_f^2 \exp\Big(-\frac{\|x-x'\|^2}{2\ell^2}\Big), \tag{1}$$

where ℓ (the length-scale) controls how rapidly correlations decay with distance, and σ_f^2 sets the overall signal variance. By fitting ℓ, σ_f, and the observation noise σ_n^2 via cross-validation, GP regression balances data fidelity and smoothness, yielding both accurate predictions and principled uncertainty estimates.

Tree Based Ensembles. In this work, tree-based ensembles are included as a practical, ensemble-based baseline for uncertainty estimation. Rather than providing a fully probabilistic characterization of uncertainty, these models approximate predictive uncertainty through the dispersion of predictions across multiple learners, a strategy that is widely adopted in practice due to its scalability, robustness, and strong performance on tabular data. Boosting [34] is an ensemble learning paradigm that combines multiple weak learners into a single strong predictor by sequentially fitting each new learner to the residual errors of the aggregate model. At each iteration t, a base estimator $f_t(x)$ is trained to minimize a loss $\ell\big(y, \hat{y}^{(t-1)} + f_t(x)\big)$, and the ensemble prediction is updated as

$$\hat{y}^{(t)}(x) = \hat{y}^{(t-1)}(x) + \eta\, f_t(x), \tag{2}$$

where η is a learning rate controlling the contribution of each learner. Common choices for base learners are shallow decision trees. Predictive uncertainty can be empirically estimated by computing the variance of the individual base learners' predictions for the same input. A small variance indicates strong agreement among learners (low uncertainty), whereas a large variance reflects disagreement (high uncertainty).

Popular implementations of boosting include XGBoost [35] and LightGBM [36]. In this work we use XGBoost due to its efficiency and built-in regularization, configuring each tree with row (s_r) and column subsampling (s_c) to promote model diversity. By tuning the subsampling rates s_r and s_c, we balance the trade-off between predictive accuracy and uncertainty calibration, ensuring that the ensemble both performs well and provides meaningful variance estimates.

Monte Carlo Dropout. MC-Dropout [11] interprets standard dropout regularization in deep neural networks as approximate variational inference. During training, each hidden unit is randomly masked with probability p via dropout, which can be seen as drawing from a variational posterior. At test time, we keep the dropout mechanism active: for each of the T stochastic forward passes through the network, hidden units are randomly deactivated with probability p. As a result, each pass effectively samples a different subnetwork, yielding predictive samples $\{f_t(x)\}_{t=1}^{T}$. The predictive mean and variance are then approximated as

$$\hat{\mu}(x) = \frac{1}{T}\sum_{t=1}^{T} f_t(x), \qquad \widehat{\mathrm{Var}}(x) = \frac{1}{T-1}\sum_{t=1}^{T}\big(f_t(x) - \hat{\mu}(x)\big)^2. \tag{3}$$

Here, $\hat{\mu}(x)$ serves as the point-estimate and $\widehat{\mathrm{Var}}(x)$ quantifies predictive uncertainty. MC-Dropout can also scale to large datasets, though it requires T evaluations per input to form a reliable uncertainty estimate.

3.2. Synthetic data generation techniques

To augment the dataset in the most informative regions identified through uncertainty estimation, we evaluate the effectiveness of three distinct generative modeling techniques: CTGANs, VAEs, and LLMs. Each of these techniques provides a mechanism for producing synthetic samples that capture different aspects of the underlying data distribution. Below we summarize their main principles.

Conditional Tabular GANs. Conditional Tabular GANs (CTGANs) [37] are GANs specifically designed to handle tabular data with complex distributions and mixed data types. Denote a data record as $\mathbf{x} = [x^{(1)}, \ldots, x^{(p)}]$, with continuous indices $\mathcal{N}$ and categorical indices C. For each continuous column $j \in \mathcal{N}$, CTGAN fits a Gaussian mixture model of the form

$$x^{(j)} \sim \sum_{k=1}^{K_j} \pi_{j,k}\, \mathcal{N}(\mu_{j,k}, \sigma_{j,k}^2), \tag{4}$$

using expectation–maximization to estimate weights $\{\pi_{j,k}\}$, means $\{\mu_{j,k}\}$, and variances $\{\sigma_{j,k}^2\}$. During both training and sampling, a component k is drawn with probability $\pi_{j,k}$ and the corresponding value is normalized via

$$\tilde{x}^{(j)} = \frac{x^{(j)} - \mu_{j,k}}{\sigma_{j,k}}, \tag{5}$$

ensuring each continuous feature appears approximately Gaussian to the network. For each categorical column $j \in C$ with m_j categories, the value is one-hot encoded into $\mathbf{c}^{(j)} \in \{0,1\}^{m_j}$. A conditional vector $\mathbf{c}$ is constructed by sampling a column j^* and category index k^* according to their empirical frequencies, setting $\mathbf{c}^{(j^*)} = \mathbf{e}_{k^*}$ (the standard basis vector) and zeroing all other slots in $\mathbf{c}$.

The generator G then takes as input the concatenation $[\mathbf{z}; \mathbf{c}]$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and produces $\tilde{\mathbf{x}} = G([\mathbf{z}; \mathbf{c}])$, outputting normalized continuous values and categorical logits. The discriminator D receives $[\tilde{\mathbf{x}}; \mathbf{c}]$ and returns a scalar score indicating real versus fake under the same conditional vector. Training minimizes a Wasserstein GAN objective with gradient penalty. Over a minibatch of size m, the discriminator loss is

$$\mathcal{L}_D = \frac{1}{m}\sum_{i=1}^{m}\big[D(\mathbf{x}'^{(i)}) - D(\mathbf{x}^{(i)})\big] + \lambda\, \mathbb{E}_{\hat{\mathbf{x}}}\Big(\|\nabla_{\hat{\mathbf{x}}} D([\hat{\mathbf{x}}; \mathbf{c}])\|_2 - 1\Big)^2, \tag{6}$$

where $\{\mathbf{x}^{(i)}\}$ are real samples and $\{\mathbf{x}'^{(i)}\}$ are generated samples. The generator loss is

$$\mathcal{L}_G = -\frac{1}{m}\sum_{i=1}^{m} D(\mathbf{x}'^{(i)}) + H, \tag{7}$$

where H denotes the conditional-vector entropy term that encourages the generated sample $\mathbf{x}'$ to satisfy the chosen category k^* in column j^*.

Variational Autoencoders. VAEs [14] are latent-variable models that learn an approximate posterior $q_\phi(z \mid x)$ parameterized by an encoder network, which outputs a mean $\mu_\phi(x)$ and standard deviation $\sigma_\phi(x)$. A latent sample z is drawn via the reparameterization trick, $z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. The decoder network $p_\theta(x \mid z)$ then reconstructs $\tilde{x}$ from z, and the model is trained by maximizing the evidence lower bound (ELBO), combining a reconstruction term (e.g., log-likelihood) and a KL divergence $\mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))$. This probabilistic framework enables the VAE to capture underlying data structure and generate diverse samples even with limited real data.

Large Language Models. LLMs, based on transformer architectures, leverage multi-head self-attention to capture complex dependencies across tokens. By fine-tuning an LLM on prompts that encode column names and cell values, the model learns to generate tabular rows consistent with input patterns [38]. Specifically, we adapt a pretrained transformer by supplying structured prompts and train it to predict missing entries, thereby learning conditional distributions over mixed-type features. While computationally more demanding, LLMs offer a high flexibility and semantic coherence, which make them a data augmentation approach worth considering.

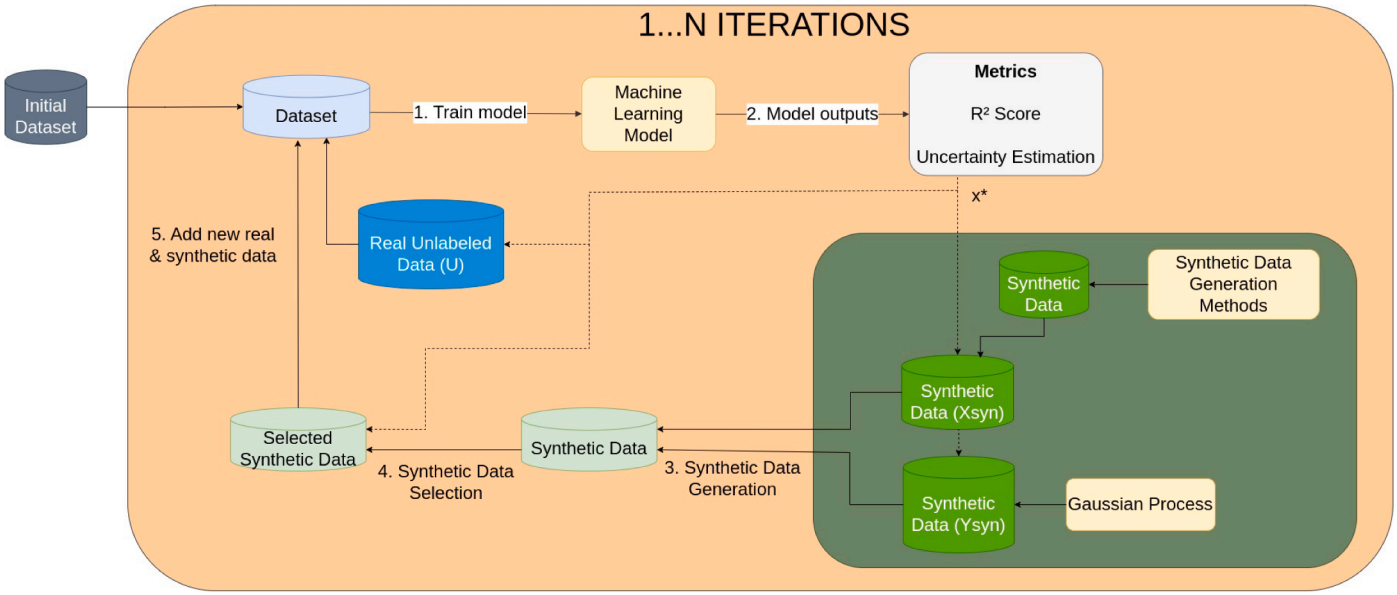


Fig. 1. Overview of the proposed uncertainty-guided synthetic data augmentation framework. The predictive model is first trained, and the most uncertain real sample is selected based on model uncertainty. A generative model is then used to produce plausible candidate input configurations, while synthetic labels are assigned using the uncertainty estimation model to ensure consistency and robustness in low-data regimes. Selected synthetic samples and the newly acquired real sample are added to the dataset, and the model is retrained iteratively.

3.3. Proposed methodology

In this work, we propose to combine predictive uncertainty estimation with data augmentation in a unified active learning pipeline. Specifically, uncertainty serves as the criterion to select the next informative real sample to be labeled, while generative models provide additional synthetic samples at each iteration, so that the model does not only learn from one new point at a time. This combined strategy aims to accelerate uncertainty reduction and improve data efficiency. The overall methodology is summarized in Fig. 1.

The methodology consists of the following steps:

1. **Model Training (Initial Stage)**: The process begins by training a predictive model which is able to provide not only point predictions but also an estimate of predictive uncertainty on an initial dataset. Although our experiments focus on GP + VAE (selected because they yielded the best results in our evaluation), the framework is agnostic to the specific uncertainty estimator and generative model used.
2. **Uncertainty Estimation and Region Selection**: After training, the model outputs both performance metrics (e.g., R^2 score) and uncertainty estimates across the input space. Let $\mathcal{U}$ denote the pool of unlabeled candidate inputs. Using the GP's posterior uncertainty, we identify the unlabeled input x^* with highest predictive variance,

$$x^* = \arg\max_{x \in \mathcal{U}} \mathrm{Var}[f(x)],$$

and query its true label $y^* = f(x^*)$, adding (x^*, y^*) to the training set.
3. **Synthetic Data Generation**: Generate synthetic candidates X_{syn} in the neighborhood of x^* using a chosen generator (LLM, VAE, CT-GAN, etc.). The role of the generative model at this stage is not to provide synthetic labels, but to generate candidate input configurations that are statistically consistent with the underlying data distribution. By learning the joint distribution of the input variables, the generator constrains the sampling process to a plausible data manifold, preserving correlations and structural dependencies that would generally be ignored by random sampling around x^*.

   Given that the synthetic data generator may not reliably capture the input-output mapping early in training, as observed in preliminary empirical analyses, we avoid using its generated labels directly. Instead, we employ the current GP model to pseudo-label each synthetic sample x_{syn} using its GP predictive mean

$$y_{\mathrm{syn}} = \mathbb{E}\big[f(x_{\mathrm{syn}})\big],$$

   which provides a conservative and consistent labeling strategy and mitigates the risk of propagating incorrect synthetic labels, particularly in low-data regimes. Moreover, since the uncertainty estimation model is iteratively updated with informative real samples via active learning, the quality of pseudo-labels progressively improves over successive iterations, reducing potential bias over time.

   Overall, within the active-learning framework adopted in this work, the generative model is used to propose plausible candidate inputs, while the uncertainty estimation model acts as an oracle responsible for evaluating and labeling these candidates in a consistent manner.
4. **Synthetic Data Selection**: The newly generated synthetic data is then evaluated to determine its quality and its contribution to improving model performance. The L1 distance between each x_{syn} and x^* is computed, and then the top 5 nearest synthetics $\{(x_{\mathrm{syn}}, y_{\mathrm{syn}})\}$ are selected for augmentation.
5. **Add New Data (Real & Synthetic)**: Append the chosen 5 synthetics together with the newly acquired real point to the training set. Retrain the GP on this extended dataset.

The process can then be iterated, progressively improving both the model performance and the quality of data generation over successive loops.

4. Experimental setup

This section details the experimental conditions under which the methods described in Section 3 were evaluated. We first introduce the real-world femtosecond laser texturing dataset used in our study (Section 4.1), followed by the hyperparameter configurations adopted for both uncertainty estimators and generative models. (Section 4.2). We then describe the evaluation metrics employed to assess performance of both data augmentation methods and active learning models (Section 4.3).

Evaluated strategies and generative model evaluation. In our experiments, we benchmarked three uncertainty-based acquisition strategies corresponding to the uncertainty estimation methods described in Section 3.1: GPs, Tree-Based Ensembles, and MC-Dropout. In addition, we consider a naive random acquisition strategy as a baseline, where new points are selected uniformly at random from the unlabeled pool. To allow fair comparison, we still fit a GP on the labeled set to track posterior variance. This allows us to observe how uncertainty evolves when points are added without any selection criterion. Comparing this random acquisition to our uncertainty-driven approach lets us quantify the true benefit of informed selection over purely random sampling.

On the other hand, each generative model (CTGAN, VAE, LLM) was tested independently to assess its ability to generate high-quality synthetic data in data-scarce scenarios and to improve overall model performance. Rather than combining these approaches, the objective was to systematically analyze their strengths, limitations, and suitability within the context of uncertainty-guided experimental design.

4.1. Dataset

We evaluate our methodology on a real industrial dataset from a femtosecond-laser texturing process [39,40]. Femtosecond laser surface texturing is gaining increased interest for optimizing tribological behaviour. However, searching the optimum parameter combination becomes a tedious activity in laser processing, since multitude potential process parameter combinations exist [41,42], and its selection is often conducted through time-consuming trial-and-error processes, resulting in comprised DoEs. In our case study, among all the femtosecond laser process parameters, laser fluence, scanning speed and pulse repetition rate were analysed, as they are considered critical [43]. Specifically, our dataset comprises $N = 192$ observations, where each sample includes those three input features corresponding to laser parameters:

- x_1: Pulse energy (μJ)
- x_2: Repetition rate (kHz)
- x_3: Scan speed (mm/s)

and one target variable:

- y: Texture depth (μm)

The use case goal is texture depth prediction, driven by the pivotal role that lubrication film thickness plays in different tribology applications [44]. Due to the nature of the underlying physical process and experimental setting, the dataset exhibits intrinsic variability in the observed outputs.

Train-Test Split. For performing our active learning methodology, we randomly select $n_{\text{initial}} = 20$ samples as the initial labeled set; the remaining 172 samples form the unlabeled pool. In addition, we reserve a fully independent test set of 48 observations (20% of the total) to evaluate final predictive performance. This split ensures that the test set remains unseen throughout model training and hyperparameter tuning (Section 4.2), providing an unbiased estimate of generalization.

Note: to assess statistical variance, all experiments are repeated over $n_{\text{runs}} = 10$ independent trials, each with a different random seed for the initial labeled set.

4.2. Hyper-parameter selection

Although in a fully deployed scenario one might only have access to a small initial batch of data, here we optimize all uncertainty-based model hyperparameters via 5-fold cross-validation (CV) over the entire training set. For the MC-Dropout network, where a full 5-fold CV would require retraining for multiple epochs and performing $T = 50$ stochastic passes per fold, we instead used a single 20% hold-out split to select the hyperparameters. Our objective is not to perform exhaustive hyperparameter optimization for each predictive model, but to ensure a fair and interpretable comparison between acquisition strategies under consistent conditions.

Table 1
Grid of hyperparameters tested for each uncertainty estimation method. Selected hyperparameters are highlighted in **bold**.

Method	Hyperparameter	Tested Values
GP	Kernel constant (σ_f^2)	{0.1, **1.0**, 10.0}
	Kernel length-scale (ℓ)	{0.5, **1.0**, 2.0}
	Alpha (α)	{1e-10, 1e-5, **1e-2**}
XGBoost Ensemble	# of estimators	{50, **100**, 200}
	Max tree depth	{**3**, 6, 9}
	Row subsample	{**0.6**, 0.8}
	Column subsample (by tree)	{0.6, **0.8**}
MC-Dropout	Dropout rate	{**0.1**, 0.2, 0.3}
	Learning rate	{**1e-3**, 1e-4}
	Number of epochs	{50, **100**}

Table 2
Hyperparameter grids evaluated for each synthetic data generation technique.

Method	Hyperparameter	Values Tested
CTGAN	batch_size	10, 30, 100, 500
	generator_lr / discriminator_lr	1e-3, 1e-4
VAE	batch_size	10, 30, 100, 500
	embedding_dim	10, 20, 50
LLM	batch_size	8, 16, 32, 64
	temperature	0.3, 0.6, 0.9

The tested hyperparameter values are intentionally limited but representative: they span distinct regimes of model complexity and regularization (e.g., multiple orders of magnitude for GP regularization/noise), and reflect commonly used ranges for each model family. This design choice avoids confounding the comparison with aggressive method-specific tuning, and ensures that subsequent performance differences arise primarily from the acquisition strategies rather than from disparate optimization effort. This results in a total of 27, 36, and 12 evaluated configurations for GP, XGBoost, and MC-Dropout, respectively. Table 1 shows the tested hyperparameter; those highlighted in bold yielded the best results and were used in the experiments.

Similarly, Table 2 shows the hyperparameters tested for the synthetic generation methods. In order to identify the most effective configuration for each generative technique under different levels of data availability, we trained each generator (CTGAN, VAE, LLM) on subsets corresponding to 10%, 25%, 50%, 75%, and 100% of the real training set, while fixing the number of training epochs of CTGAN, VAE, and LLM to 500, 500, and 300, respectively. We then performed a structured grid search over all combinations of the listed hyperparameters for each augmentation technique at each training fraction and selected the configuration yielding the highest mean test-set R^2 (see Table 3). The performance obtained with these settings is reported in Section 5.2.

For our LLM-driven augmentation, we fine-tuned the lightweight DistilGPT-2 model [45], which retains the core transformer architecture of GPT-2 in a compressed form. DistilGPT-2 comprises approximately 82 million parameters distributed across six transformer blocks, each featuring 12 self-attention heads and an embedding dimension of 768. This configuration offers a context window of up to 1024 tokens, balancing representational capacity with computational efficiency.

4.3. Evaluation metrics

To assess the effectiveness of our proposed methodology, we consider two complementary categories of evaluation metrics: *1) uncertainty estimation metrics* and *2) synthetic data quality metrics*.

Table 3
Optimal hyperparameter settings chosen for each synthetic-data technique at different data fractions.

Method	% Data	batch_size	lr/emb_dim/temp
CTGAN	10%	100	1e-3
	25%	100	1e-3
	50%	500	1e-3
	75%	30	1e-3
	100%	500	1e-3
VAE	10%	10	emb_dim = 20
	25%	10	emb_dim = 10
	50%	10	emb_dim = 10
	75%	10	emb_dim = 10
	100%	30	emb_dim = 20
LLM	10%	8	temp = 0.9
	25%	8	temp = 0.6
	50%	8	temp = 0.3
	75%	8	temp = 0.6
	100%	8	temp = 0.3

1) Uncertainty Estimation Metrics. To quantify how well each acquisition strategy reduces total uncertainty and improves predictive accuracy, we record at each active learning iteration:

- **Test-set** R^2 (↑): the coefficient of determination quantifies the proportion of variance in the target variable that is explained by the model over the test data. Higher R^2 values indicate better model performance and improved predictive power.
- **Mean predictive uncertainty** (↓): the average posterior standard deviation (for GP and ensemble) or MC-Dropout estimate over the pool of unlabeled points.
- **Mean Absolute Error** (↓): the mean absolute error (MAE) is defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|,$$

and provides an absolute error-based measure of predictive accuracy in the original scale of the target variable.

2) Synthetic Data Quality Metrics. To assess the fidelity and utility of the synthetic data generated by each method (CTGAN, VAE, and LLM), we compute four complementary metrics.

- **Train-on-Synthetic, Test-on-Real (TSTR):** We train an XGBoost regressor on each synthetic dataset and evaluate its performance on the held-out real test set. We report the coefficient of determination R^2 on the real test samples. A high R^2 indicates that the synthetic data captures the input-output mapping of the real process.
- **Distance to Closest Record (DCR):** To measure manifold coverage, we fit a 1-nearest-neighbor model on the entire real training set and compute, for each synthetic sample, the Manhattan distance to its nearest real neighbor in the original feature space. We then examine the distribution of these distances. Small median and narrow spread imply that synthetic points lie close to the true data manifold without creating large state spaces without being covered.
- **Discriminator Score:** We quantify how easily a classifier can distinguish real from synthetic samples. We label real training points as 0 and synthetic points as 1, concatenate them, and train a Random Forest classifier with 5-fold cross-validation, reporting the accuracy of the classifier. An accuracy near 0.5 indicates that the classifier cannot reliably tell real and synthetic apart, implying high generative fidelity.
- **Bivariate Joint Distribution Plots:** As a qualitative check on feature dependencies, we generate scatter plots overlaying real and synthetic points for the three most critical input-output pairs. By visually inspecting these joint distributions, we verify that each method preserves the shape, range, and correlation structure of the original data.

5. Results and discussion

This section presents the experimental results and analysis of our proposed methodology. First, we evaluate the performance of the different uncertainty estimation techniques (Section 5.1). Next, the quality and effectiveness of the synthetic data generated by different generative models is analyzed (Section 5.2). Lastly, we provide the outcomes considering the pipeline of both uncertainty estimators and generative models when applied iteratively in an active learning setup (Section 5.3).

5.1. Uncertainty measurement analysis

In this section, we present the results of the uncertainty methods presented in Section 3.1.

Fig. 2 shows how the mean predictive uncertainty evolves over the 20 acquisition steps for each sampling strategy. The XGBoost ensemble stands out by driving uncertainty down most sharply from about 0.33 at iteration 1 to 0.14 at iteration 20, showing a reduction of 57% and reflecting its ability to rapidly resolve high-variance regions through diverse boosted trees. The Gaussian-process sampler also decreases uncertainty, from 0.466 to 0.27 (42.1% reduction) as its exact posterior variance concentrates around observed data. Random sampling, although being blind to model confidence, still reduces uncertainty from 0.466 to 0.33 (27.7%), simply by filling unexplored regions. However, we can see how it exhibits intermittent increases in uncertainty when, as it may select points in already well-covered areas, leaving truly uncertain regions untouched. In contrast, MC-Dropout is not able to reduce the uncertainty over the iterations, even increasing it from 0.44 to 0.46 and suggesting

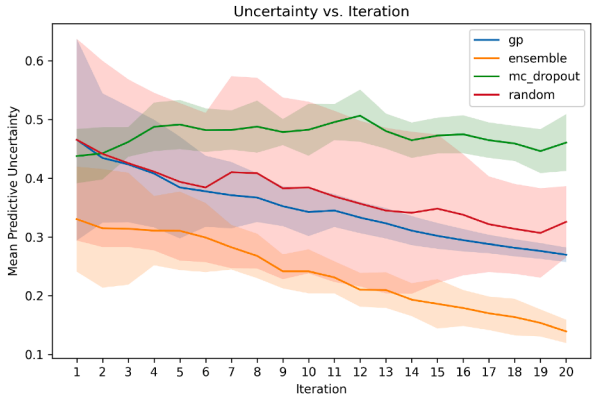


Fig. 2. Mean predictive uncertainty for GP, XGBoost ensemble, MC-Dropout, and random sampling over 20 iterations. The shadowed area represents the standard deviation of the results performed over 10 independent runs.

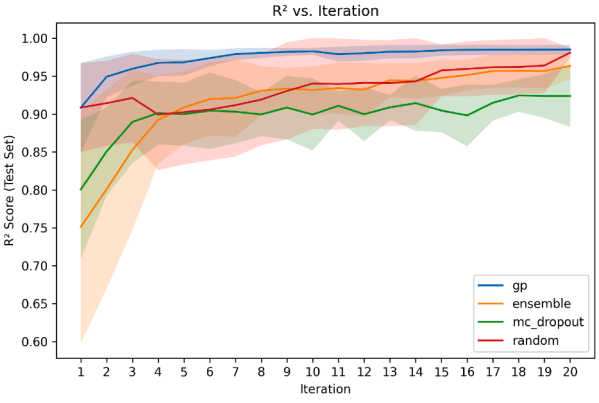


Fig. 3. Mean test-set R^2 score for GP, XGBoost ensemble, MC-Dropout, and random sampling over 20 iterations. The shadowed area represents the standard deviation of the results performed over 10 independent runs.

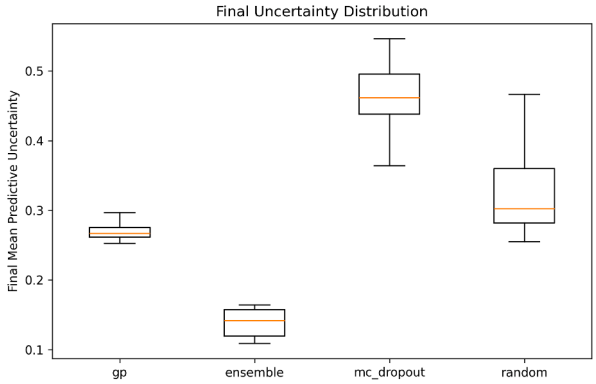


Fig. 4. Boxplots of final mean predictive uncertainty after 20 iterations, aggregated over 10 independent runs, for each sampling method.

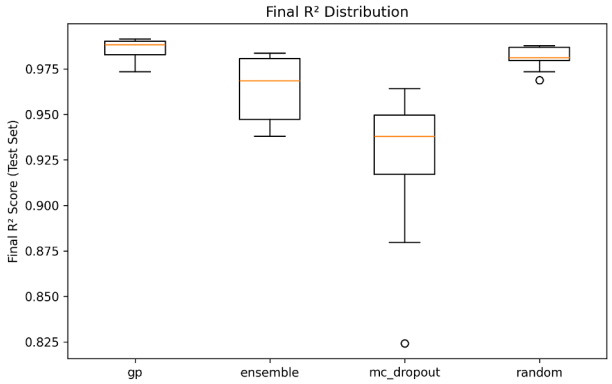


Fig. 5. Boxplots of final test-set R^2 scores after 20 iterations, aggregated over 10 independent runs, for each sampling method.

that dropout-based variance can overestimate model doubt when new samples fail to improve confidence. The width of the shaded bands further reinforces these differences: GP and ensemble curves have narrower bands, indicating consistent behavior across runs, whereas random sampling exhibit much wider variability, underscoring their less consistent uncertainty estimates.

Fig. 3 depicts how test-set R^2 score evolves over the same iteration and independent runs. The GP sampler delivers the best accuracy results, climbing from 0.91 on iteration 1 to 0.98 by iteration 7, reaching a plateau near 0.99 thereafter. The XGBoost ensemble follows a steadier ascent, rising from 0.75 to 0.96 by iteration 20, but not arriving to the performance shown by GP. Random sampling, despite its possible uninformative acquisitions, achieves comparable final accuracy of uncertainty based GPs, but more slowly and with greater variability. Finally, MC-Dropout lags behind, improving from 0.80 to 0.92 over all iterations.

To complement the R^2 analysis, MAE results are reported in Appendix A. The MAE curves and final distributions exhibit consistent trends across all strategies and do not alter the conclusions drawn from the main results.

Taken together, Figs. 2 and 3 confirm an inverse relationship between total uncertainty and predictive performance: methods that rapidly drive down uncertainty (GP, ensemble) also yield the fastest and most stable R^2 improvements, whereas those that maintain or even increase uncertainty (MC-Dropout, random) see slower or uneven accuracy gains. This alignment underscores that effective uncertainty reduction is key to maximizing learning efficiency under a fixed sampling budget. Interestingly, random sampling achieves a final R^2 comparable to the ensemble despite the absence of an uncertainty-driven criterion. However, this performance comes at the cost of greater variability across runs.

Fig. 4 presents boxplots of the final mean predictive uncertainty (after 20 iterations) across 10 independent runs. The XGBoost ensemble achieves the lowest median uncertainty, reflecting its rapid collapse of model-variance. The Gaussian-process sampler follows with very little spread (tightest interquartile range), illustrating its consistent contraction of true variance. Random sampling ends with a much wider whisker span, confirming the erratic uncertainty spikes observed earlier. Finally, MC-Dropout sits highest and also shows broad variability run-to-run, being consistent with wider predictive uncertainty in dropout-based neural nets.

Finally, Fig. 5 displays boxplots of the final test-set R^2 scores across the same runs. The GP sampler not only achieves the highest median R^2 but does so with minimal dispersion, indicating both strong accuracy and low run-to-run variability. Random sampling follows closely but with wider spread, reflecting its less targeted data selection. The XGBoost ensemble attains moderately high R^2 scores yet exhibits greater dispersion than random, consistent with its overconfident uncertainty estimates that sometimes miss truly informative points. MC-Dropout shows the lowest and most variable R^2 scores performance, mirroring its failure to reduce uncertainty effectively.

Based on our comprehensive comparison, the Gaussian-process sampler emerges as the clear best choice for our application. It consistently delivers the most faithful uncertainty estimates, reflected in both steadily contracting variance and narrow run-to-run bands, and this reliable uncertainty guidance translates directly into the highest and most stable test-set R^2 score. While the XGBoost ensemble aggressively collapses its own heuristic variance, it does so at the expense of masking systematic errors, yielding lower accuracy. Random sampling achieves comparable accuracy only after many more draws and with greater variability, and MC-Dropout neither reduces uncertainty effectively nor drives accuracy gains. Therefore, for our femtosecond-laser texturing use case, we adopt the Gaussian-process acquisition strategy as the sole uncertainty-guided sampler in our full hybrid methodology.

5.2. Synthetic data generation analysis

After evaluating how uncertainty-guided acquisition strategies affect model confidence and accuracy, we now turn to the complementary task of synthesizing new training samples in those uncertain regions. This section evaluates the performance of the synthetic data generation techniques tested in our methodology: CTGAN, VAEs and LLMs. Each generative method is assessed independently using the evaluation metrics presented in Section 4.3.

5.2.1. Train on synthetic, test on real

We begin by evaluating whether synthetic samples alone can support predictive modeling on real data. This “train on synthetic, test on real” setup provides a functional benchmark for the utility of each generator.

Table 4 reports the mean test-set R^2 (±std) of an XGBoost regressor trained exclusively on synthetic datasets. Each generative model (CTGAN, VAE, LLM) was trained on a fraction of the real training data (10%, 25%, 50%, 75%, or 100%) and then used to produce a synthetic dataset matching the size of the full real training set (192 samples in our case). The downstream XGBoost model was fitted on these synthetic samples and evaluated on the held-out real test set. We adopt XGBoost as the downstream predictor since boosting algorithms remain among the most robust and widely used approaches for tabular data [46,47].

In our study, CTGAN delivers very poor predictive utility at all data budgets, never exceeding $R^2 \approx 0.40$ even with 50% of the real data and falling below 0.10 at 25% and 100%. Notably, CTGAN performance does not improve when trained with larger fractions of the dataset. With the full set of 192 real samples used to train, the generated samples yield a very low $R^2 \approx 0.04$, suggesting that the model struggles to capture the input-output relationships required for downstream prediction. This can be consistent with known limitations of GANs with small datasets,

Table 4
Mean test-set $R^2 \pm$ std after training a XGBoost with solely synthetic samples. Synthetic datasets were generated by first training each generator (CTGAN, VAE, LLM) on a given fraction of the real training data (10%, 25%, 50%, 75%, 100%), then sampling the same number of synthetic points as the original real-training set, 192 data points in our case. The downstream model is evaluated on the real hold-out test set. "Original" column reports the performance when training on the full real dataset. Best results are highlighted in **bold**.

Percentage	CTGAN	VAE	LLM	Original
10%	0.30 ± 0.07	0.72 ± 0.12	-0.14 ± 0.46	–
25%	0.12 ± 0.12	0.77 ± 0.07	0.26 ± 0.21	–
50%	0.39 ± 0.24	0.88 ± 0.02	0.71 ± 0.01	–
75%	0.14 ± 0.40	0.87 ± 0.05	0.83 ± 0.09	–
100%	0.04 ± 0.09	0.91 ± 0.04	0.92 ± 0.01	0.98

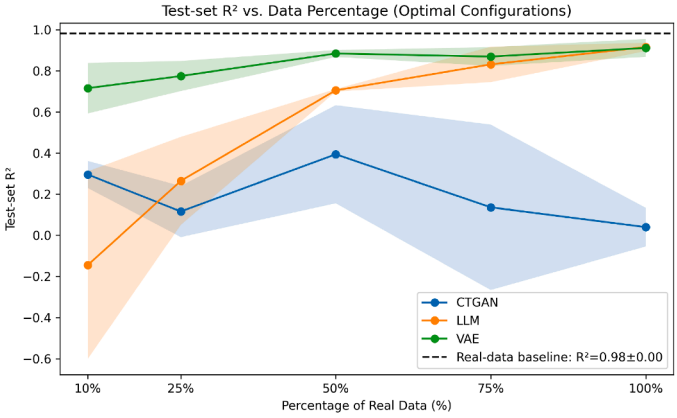


Fig. 6. Test-set R^2 of an XGBoost regressor trained on synthetic datasets generated by CTGAN, VAE, and LLM with their optimal hyperparameters. Synthetic datasets were generated by training each generator (CTGAN, VAE, LLM) on a given fraction of the real training data (10%, 25%, 50%, 75%, 100%). Shaded bands represent the variability over three independent runs, and the dashed horizontal line indicates the real-data baseline performance.

where issues such as mode collapse and poor conditional modeling can lead to synthetic data that lacks predictive value, as the discriminator overfits rapidly and the generator fails to learn diverse or informative patterns [20].

By contrast, the VAE is markedly more robust under data scarcity, achieving $R^2 = 0.72 \pm 0.12$ with only 10% of the data and steadily improving to $R^2 = 0.91 \pm 0.04$ at full data. The LLM exhibits the opposite profile: it fails to capture the mapping when data are very limited ($R^2 = -0.14 \pm 0.46$ at 10%), but performance rapidly improves with more examples, surpassing CTGAN by 25% and approaching VAE levels by 100% ($R^2 = 0.92 \pm 0.02$).

These numerical trends shown in Table 4 are illustrated in Fig. 6, which plots the R^2 against the fraction of real data for each method, further confirming the numerical trends reported in Table 4. These results shows that under extreme data scarcity the VAE's inductive bias yields the most reliable synthetic samples for regression, whereas the LLM requires a moderate amount of data (> 25%) to generalize effectively. CTGAN, in our experiments, fails to produce useful examples at any scale.

5.2.2. Distance to closest record

In order to quantify how closely synthetic samples are to the real data manifold, we computed the L1-based DCR. For each synthetic point x_{syn}, DCR is defined as the L1 distance to its nearest neighbor in the real training set. As a real-data baseline, we compute the DCR of each real test point with respect to its nearest neighbor in the real training set, yielding a reference median of 2.2 units.

Table 5
Median L_1 DCR (units) for each synthetic method and data fraction.

Data Fraction	Baseline (Real)	CTGAN	VAE	LLM
10%	2.2	20.48	23.54	0.00
25%	2.2	21.48	19.56	0.00
50%	2.2	21.45	15.45	0.00
75%	2.2	16.63	5.09	0.00
100%	2.2	11.46	3.80	0.00

These results are shown for one single seed; corresponding analyses for the other random seeds are provided in Appendix B. Fig. 7 presents the DCR histograms for CTGAN, VAE, and LLM across five levels of data availability used to train the generators (10%, 25%, 50%, 75%, 100% of the real training set), and the bottom row reports the real Test-Train baseline. Lower DCR values indicate synthetic points closer to genuine data in feature space. Table 5 summarizes the median DCR values for each generator method and the real-data baseline (median = 2.2 units).

The real-test baseline DCR of 2.2 units shows that unseen genuine samples lie very close to the training manifold, with most distances clustered around 2–4 units and only a thin tail of outliers. By contrast, CTGAN's DCR histograms remain broad and heavily skewed toward large distances (medians between 11.5 and 21.5 units), even at 100% data, underscoring its failure to place synthetic points near any real examples. The VAE initially produces wide distributions under extreme scarcity (median $\approx$ 23.5 at 10%) but its histograms narrow sharply by 75% and 100%, matching the baseline shape (median $\approx$ 3.8), which demonstrates the VAE's progressive learning of the true feature manifold. The LLM, however, shows a spike at zero DCR across all fractions-its median is exactly 0-suggesting that, under the chosen seed, the model may be memorizing and reproducing training records rather than generating novel samples. This behavior is undesirable in the context of data augmentation, as it limits the model's ability to enrich the input space with diverse yet plausible observations.

These results confirm that only the VAE achieves both low median DCR and a histogram shape comparable to real data, striking the best balance between manifold proximity and genuine sample diversity, whereas CTGAN diverges widely and the LLM may overfit by copying.

5.2.3. Discriminator score

To assess how easily synthetic samples can be distinguished from real ones, we trained a Random Forest Classifier (label 1 = real, 0 = synthetic) on balanced training sets (144 real + 144 synthetic points) and evaluated on balanced test sets (48 real + 48 synthetic). We adopt a Random Forest Classifier for this test, as tree-based ensembles are robust on tabular data, without requiring extensive hyperparameter tuning [46]. The goal here is not to measure predictive accuracy on a regression task, but rather to quantify the "detectability" of generated data. The easier the classifier separates synthetic from genuine points (higher accuracy), the less faithful the generator. Table 6 reports the test accuracy for each generator and data fraction.

CTGAN samples remain highly distinguishable across all fractions (accuracy 0.83–0.87), with the classifier being most confident at 10% and 100% data. VAE outputs exhibit a similar level of distinguishabil-

Table 6
Discriminator score (acccuracy ± std) for each synthesis method and data fraction.

Percentage / Method	CTGAN	VAE	LLM
10%	0.87 ± 0.03	0.9 ± 0.02	0.76 ± 0.08
25%	0.85 ± 0.05	0.83 ± 0.03	0.64 ± 0.06
50%	0.86 ± 0.05	0.83 ± 0.03	0.64 ± 0.02
75%	0.83 ± 0.06	0.84 ± 0.03	0.6 ± 0.04
100%	0.87 ± 0.04	0.84 ± 0.03	0.56 ± 0.08

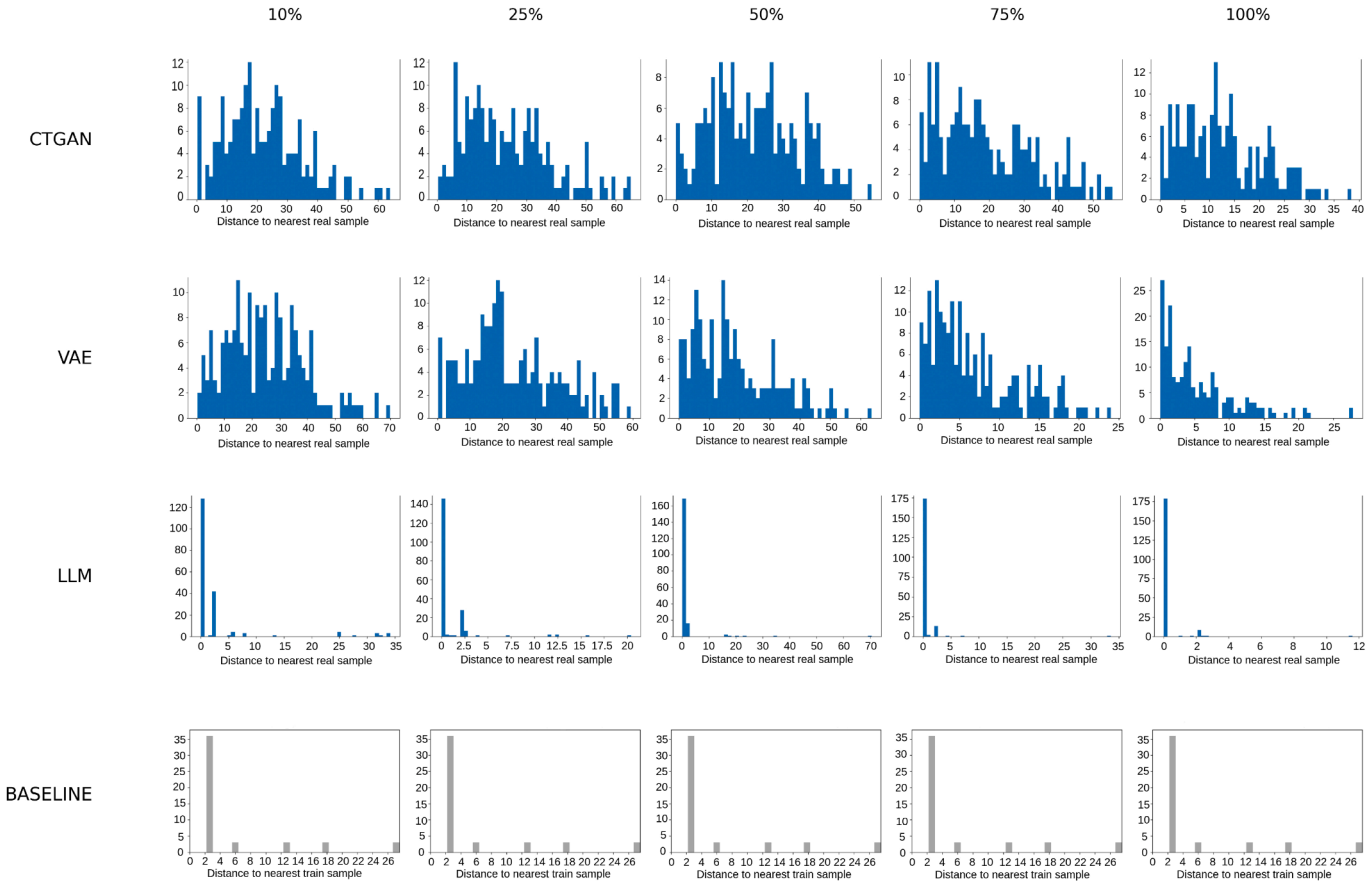


Fig. 7. L1-distance-to-closest-record (DCR) histograms for CTGAN (first row), VAE (second row), and LLM (third row) across five data-availability levels used to train the generators (10–100% of the real training set). Distances are measured to the nearest neighbor in the real training set. The bottom row shows the baseline DCR distribution for real test samples versus the training set.

ity, suggesting that VAE too leaves identifiable statistical artifacts. By contrast, the LLM's discriminator accuracy steadily declines from 0.76 at 10% to 0.56 at 100% data; with little training data it overfits, but as it ingests more examples it reproduces the real manifold so faithfully that a powerful classifier can no longer tell its samples apart from genuine ones.

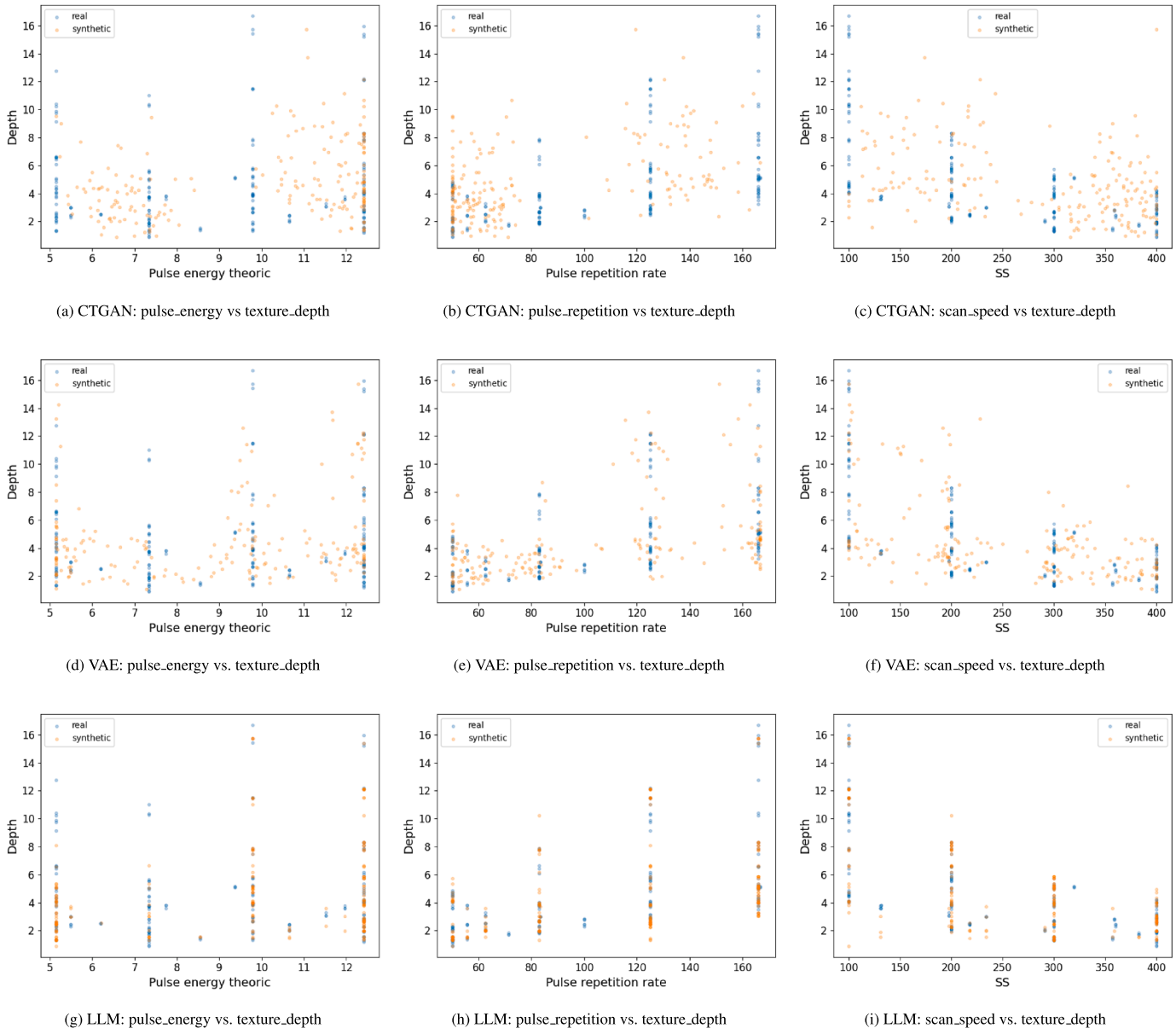
Together with our DCR and TSTR performance analyses, these findings confirm that while neither CTGAN nor VAE ever fully conceals their synthetic nature, the VAE achieves the best balance between artifact reduction and data diversity, and the LLM, given sufficient data, can generate near-indistinguishable samples. However, since our primary goal is to explore underrepresented regions of the solution space rather than simply replicate existing points, we ultimately favor generators that produce diverse coverage, even at the expense of perfect fidelity. In this scenario, the VAE remains our top choice: it progressively expands manifold coverage in previously sparse areas while avoiding the CTGAN's extreme divergence or the LLM's memorization.

5.2.4. *Bivariate joint distribution plots*

In this section, we qualitatively examine how well each generator reproduces the true bivariate relationships between key process parameters of our case study and our output variable, as mentioned in Section 4.1. Specifically, we focus on the three variable pairs of our dataset, pulse energy vs. texture depth, pulse repetition vs. texture depth and scan speed vs. texture depth. For clarity purposes, we illustrate results at the 50% data fraction setting, where each generator was trained with 50% of the available real samples. Additional results for other fractions are reported in Appendix C. Fig. 8 presents overlaid scatter plots where real samples (blue circles) and synthetic points (orange markers) are shown side-by-side for CTGAN, VAE, and LLM.

CTGAN. The CTGAN-generated synthetic data, illustrated in Figs. 8(a)–(c), reveals significant deficiencies in capturing the core experimental distributions. In Fig. 8(a), CTGAN fails to adequately represent the critical pulse energy levels at 5 μJ and 10 μJ; at the 5 μJ level, synthetic points are generated in nearby regions but miss the actual experimental cluster, while at 10 μJ, the method provides insufficient coverage of the established experimental region. Fig. 8(b) demonstrates similar shortcomings in pulse repetition rate generation, where CTGAN inadequately captures the well-defined experimental clusters around 80 kHz and 160 kHz, generating synthetic points that deviate from these critical operational frequencies. In Fig. 8(c), while CTGAN shows improved performance for scanning speed, it still exhibits poor fidelity around the 100 mm/s operational point. These systematic failures to capture established experimental regions, combined with excessive exploration in unrealistic parameter spaces, indicate that CTGAN's generation strategy prioritizes coverage over fidelity to a detrimental degree, potentially introducing synthetic samples that misrepresent the underlying physical processes.

VAE. The VAE approach, presented in Figs. 8(d)–(f), demonstrates superior performance in balancing experimental fidelity with strategic parameter space exploration. Fig. 8(d) shows that VAE successfully captures all major pulse energy levels while providing controlled exploration in their neighborhood, maintaining realistic proximity to experimental clusters without the misrepresentation observed in CTGAN. In Fig. 8(e), VAE exhibits excellent fidelity to the discrete pulse repetition



(a) CTGAN: pulse_energy vs texture_depth

(b) CTGAN: pulse_repetition vs texture_depth

(c) CTGAN: scan_speed vs texture_depth

(d) VAE: pulse_energy vs. texture_depth

(e) VAE: pulse_repetition vs. texture_depth

(f) VAE: scan_speed vs. texture_depth

(g) LLM: pulse_energy vs. texture_depth

(h) LLM: pulse_repetition vs. texture_depth

(i) LLM: scan_speed vs. texture_depth

Fig. 8. Bivariate joint distribution plots for the 50% data-fraction setting. Each generator (CTGAN, VAE, LLM) was trained with 50% of the real samples and then used to produce 192 synthetic points, matching the size of the full real dataset. The scatter plots overlay 192 real samples (blue) with 192 synthetic samples (orange) for the three key variable pairs: pulse energy vs. texture depth, pulse repetition rate vs. texture depth, and scan speed vs. texture depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

rate structure while offering appropriate exploration across the entire experimental parameter space. Fig. 8(f) further confirms VAE's balanced approach, showing comprehensive coverage of scanning speed parameters with maintained fidelity to experimental clusters. Critically, VAE avoids the excessive exploration that characterizes CTGAN, preventing the generation of unrealistic synthetic data points that could compromise model training. This controlled exploration strategy ensures that synthetic augmentation enhances parameter space coverage while preserving the physical constraints of the laser processing system.

LLM. The LLM-generated synthetic data, shown in Figs. 8(g)–(i), exhibits exceptional fidelity to real experimental data, confirming the superior performance observed in previous evaluation metrics including TSTR and DCR assessments. Fig. 8(g) demonstrates near-perfect replication of pulse energy distributions, with synthetic points precisely matching experimental clusters without deviation. Fig. 8(h) shows the same preservation of pulse repetition rate patterns, maintaining exact correspondence with established frequencies. Similarly, Fig. 8(i) reveals perfect adherence to scanning speed experimental structure. However, this high-fidelity approach comes at the cost of exploration capability, as LLM consistently fails to generate synthetic data in underrepresented parameter regions, providing minimal enhancement of parameter space coverage and limiting LLM's utility for one of the key objectives of synthetic data generation: exploring undersampled regions that could guide future experimental data or improve active learning strategies.

Together, these joint-distribution visualizations corroborate our quantitative findings: CTGAN cannot capture the full range of the real manifold, VAE strikes the best balance between reproducing correlations and exploring underrepresented zones, and the LLM achieves near-perfect fidelity at the cost of limited diversity.

5.3. Combining uncertainty estimation and synthetic data generation

Building on our earlier analysis, we now assess the full uncertainty-guided synthetic-data pipeline by using the uncertainty guided algorithm and the synthetic data generator yielding the best results: GP (Section 5.1) with VAE-based augmentation (Section 5.2).

We implement the end-to-end workflow previously described in Section 3. We begin training our GP model with an initial dataset of 20 samples. In each acquisition step (20 in total), we select the most uncertain real point according to the GP posterior and enrich the training set with VAE-generated samples drawn from the input region of highest uncertainty. We evaluate this hybrid approach on our industrial texture dataset by tracking test-set R^2 as the primary performance metric and predictive uncertainty over active-learning iterations, and by comparing final performance across four strategies: GP-based Active Learning, GP-based Active Learning with VAE augmentation, Random Sampling, and Random Sampling with VAE augmentation. Given the consistent behavior observed across performance metrics in Section 5.1, we focus on R^2 as a representative indicator for the analysis of the full pipeline. This experiment tests whether strategic VAE augmentation can accelerate uncertainty reduction and accuracy gains beyond what is achievable with real data alone.

Beyond this experiment, we further analyze other aspects to better understand the role of synthetic augmentation. First, we test extreme scenarios in which the GP is trained exclusively on synthetic points, to analyze the specific contribution of synthetics alone. Finally, we examine how the number of synthetic samples per iteration influences the trade-off between uncertainty reduction and predictive accuracy. Together, these analyses provide a comprehensive view of how synthetic augmentation impacts the dynamics of active learning.

5.3.1. Baseline comparison

Figs. 9 and 10 present the evolution of predictive performance (R^2) and mean predictive uncertainty, respectively, for four baseline strategies.

As expected, the AL strategy consistently outperforms RS in terms of both predictive accuracy and uncertainty reduction, reaching an (R^2) of ≈ 0.986 and an uncertainty of ≈ 0.27, confirming its effectiveness in targeting the most informative regions of the input space. When synthetic samples are introduced, we observe a small decrease in predictive accuracy compared to the purely real-data baselines, from approximately 0.986 to 0.974. However, this decrease is marginal and outweighed by a substantial reduction in predictive uncertainty going from ≈ 0.27 to ≈ 0.15. This effect is observed under both AL and RS, suggesting that the inclusion of synthetic data contributes to a more diverse coverage of the input space, thereby accelerating uncertainty reduction.

These findings suggest a trade-off: synthetic augmentation injects diversity into the training set, which accelerates the reduction of total uncertainty, but this comes at the cost of a small degradation in predictive performance. To better understand this balance and to quantify the influence of synthetic data more systematically, we next analyze scenarios where synthetic points play a more dominant role.

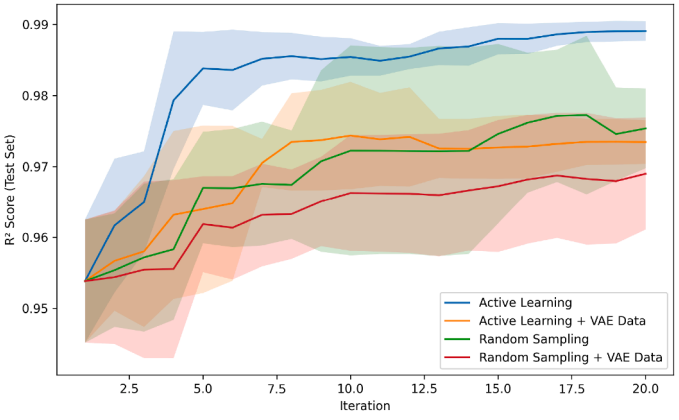


Fig. 9. R^2 performance through iterations of a GP training for different strategies. The solid lines show mean test-set R^2 across three random seeds; shaded areas indicate standard deviation across independent runs.

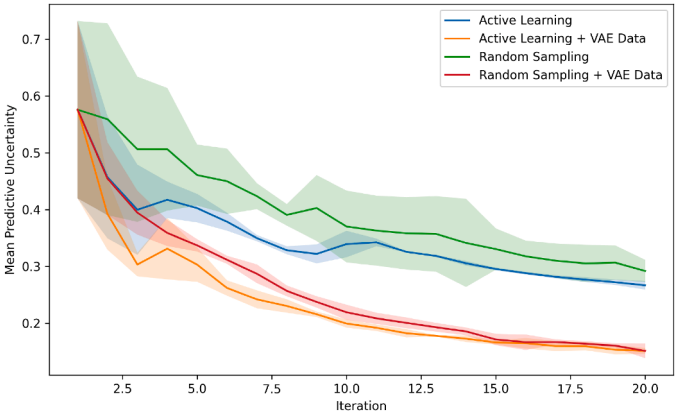


Fig. 10. Predictive uncertainty across iterations on a GP trained with different strategies. Solid curves denote mean posterior standard deviation on the test set, with standard deviation bands across independent runs.

5.3.2. Extreme scenario: Only synthetic data

To better understand the contribution of synthetic augmentation, we also consider extreme scenarios in which the GP model is updated exclusively with synthetic samples, without any new real acquisitions. We evaluate both uncertainty-guided active learning and random selection strategies for generating these synthetic points.

Results show that, in the absence of real data, predictive accuracy (R^2) remains consistently lower than in the mixed setting (Fig. 11), where synthetic points complement real acquisitions. By contrast, Fig. 12 demonstrates that uncertainty is reduced to a very similar degree across both mixed and synthetic-only settings, regardless of whether samples are chosen through active learning or random selection. This suggests that the act of injecting synthetic samples, rather than the pres-

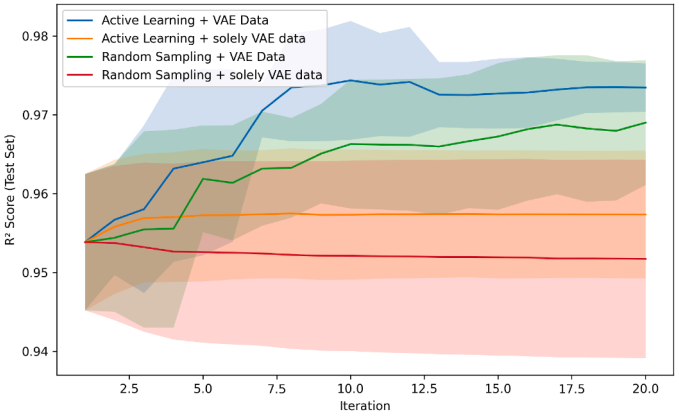


Fig. 11. Evolution of test-set R^2 across active-learning iterations for four strategies: (blue) Active Learning with real + VAE augmentation, (yellow) Active Learning with solely VAE-generated data, (green) Random Sampling with real + VAE augmentation, and (red) Random Sampling with solely VAE-generated data. Solid lines denote mean performance across three seeds; shaded bands show standard deviation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

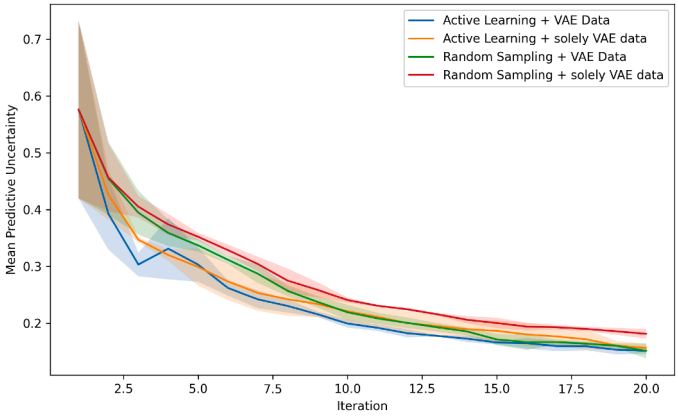


Fig. 12. Evolution of predictive uncertainty across iterations for the same four strategies as in Fig. 11. Curves report mean posterior uncertainty on the real test set, with shaded bands indicating variability across three seeds.

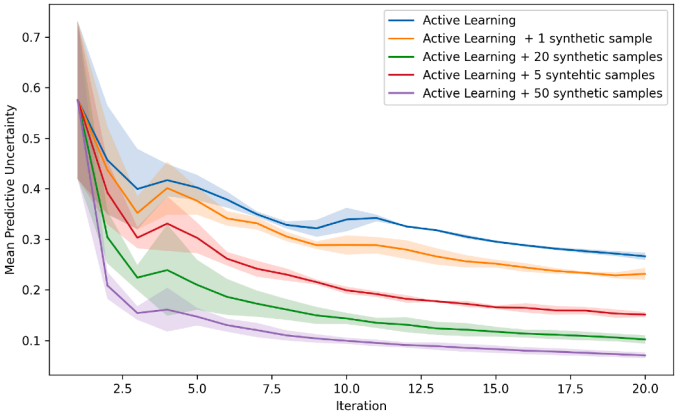


Fig. 14. Effect of the number of synthetic samples per iteration (0, 1, 5, 20, 50) under the Active Learning strategy on the evolution of mean predictive uncertainty. Solid curves denote mean posterior standard deviation on the test set, with std bands across independent runs.

ence of real acquisitions, is the main driver of accelerated uncertainty reduction. However, without real acquisitions, these gains in uncertainty come at the cost of reduced R^2, as the predictive signal becomes perturbed and increasingly biased toward the synthetic data.

Having established that synthetic samples consistently reduce predictive uncertainty but at the cost of a drop in R^2 when used without real acquisitions, the next step is to examine how the number of synthetic points influences this trade-off.

5.3.3. *Influence of number of synthetic samples*

Finally, we assess the effect of varying the number of synthetic augmentations per iteration within the Active Learning framework. To this end, we tested configurations with 0, 1, 5, 20, and 50 synthetic samples added alongside each newly acquired real point.

Results in Figs. 13 and 14 reveal a consistent pattern: introducing even a single synthetic point already accelerates uncertainty reduction compared to using real acquisitions alone. Increasing the number of synthetics further amplifies this effect, with the steepest decline in predictive uncertainty observed when adding 50 samples per iteration (Fig. 14).

However, this gain comes at the expense of predictive accuracy. As shown in Fig. 13, small additions (e.g., 1 or 5 synthetics) maintain R^2 scores closer to the real-only baseline, while larger batches (20 and especially 50) introduce stronger deviations from the underlying signal, leading to noticeable drops in predictive performance. Moreover, when 20 synthetic samples are added at each step, predictive performance collapses, and increasing the number further (e.g., 50 synthetics) does not lead to additional degradation, with the 50 synthetic data curve converging to a similarly low R^2. These findings suggest that synthetic data play a complementary role: they are most beneficial when used in moderate amounts, where they effectively guide exploration and reduce uncertainty without substantially perturbing the predictive accuracy provided by real acquisitions. In practice, this highlights the importance of carefully tuning the ratio of real to synthetic samples in order to strike an optimal balance between uncertainty reduction and predictive fidelity.

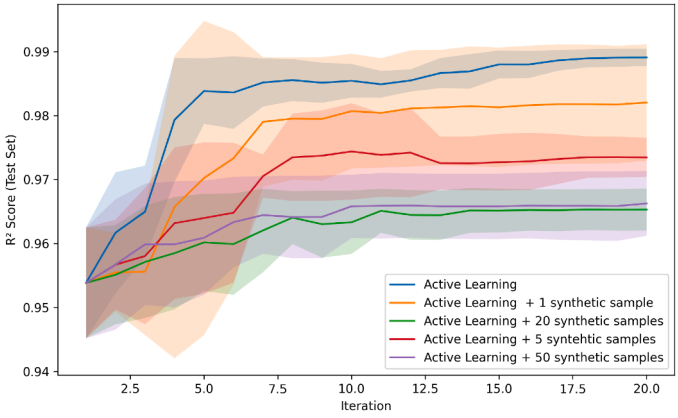


Fig. 13. Effect of the number of synthetic samples per iteration (0, 1, 5, 20, 50) under the Active Learning strategy on the evolution of predictive accuracy (R^2). The solid lines show mean test-set R^2 across three random seeds; shaded areas indicate standard deviation across independent runs.

6. Conclusion and future research

In this work, we investigated whether DoE can be guided more effectively in data-scarce scenarios by explicitly targeting regions of highest uncertainty and enriching them with synthetic samples. Using an industrial femtosecond-laser texturing dataset with only 192 real observations, we evaluated strategies that combine active learning with generative augmentation to accelerate both accuracy and confidence gains. To this end, we compared three uncertainty estimators (GP, tree-based ensembles, and MC-Dropout) and three generative models (CTGAN, VAE, and LLM), analyzing their performance both independently and in combination. In the hybrid pipeline, the best-performing uncertainty estimator and generator were integrated to iteratively select the most informative real points while augmenting each acquisition with different proportion of synthetic data.

Our results show that GP provide the most reliable balance between predictive accuracy and uncertainty reduction under extreme scarcity reaching a test-set performance of $R^2 \approx 0.98$ and reducing predictive uncertainty in a 42.1%. For synthetic data augmentation, each generator was trained using different proportion of the available real dataset (10%, 25%, 50%, 75%, and 100%), simulating different levels of data scarcity. Across these conditions, VAEs consistently produced high-utility samples, with train-on-synthetic, test-on-real R^2 increasing from ≈ 0.72 at 10% to ≈ 0.91 at 100%, while also showing realistic distributions in DCR and bivariate joint plots. Transformer-based LLMs achieved strong scores when trained with abundant data (≈ 0.92 at 100%), but they lacked diversity and frequently reproduced training records, reflecting overfitting. By contrast, CTGAN underperformed in all cases ($R^2 < 0.40$, as low as ≈ 0.04 at 100%), consistent with known instability of GANs on small tabular datasets.

By combining the two best-performing methods, GP for uncertainty estimation and VAE for synthetic generation, we integrated them into a single active learning loop. In this hybrid setting, VAE-based augmentation accelerated uncertainty reduction: the mean uncertainty decreased from ≈ 0.27 (real-only AL) to ≈ 0.15 (AL + VAE), with only a small drop in R^2 ($0.986 \rightarrow 0.975$). Synthetic-only updates also reduced uncertainty, but predictive accuracy remained consistently lower, confirming that real acquisitions are essential to anchor the model. Varying the number of synthetic data per iteration revealed a clear trade-off: 1 to 5 synthetic samples provided faster confidence gains with minimal accuracy loss, while using 20 to 50 synthetic samples perturbed the signal and caused a collapse in R^2. These findings highlight the complementary roles of real and synthetic data: real samples preserve accuracy, while synthetic ones accelerate the reduction of predictive uncertainty by enriching coverage of the input space. In cost-sensitive DoE, this trade-off is attractive when faster convergence in confidence outweighs small losses in predictive performance, validating GP + VAE as an effective strategy for scarce-data exploration.

Future Work. Looking forward, several directions could further enhance and generalize our investigation. First, exploring alternative generative paradigms such as diffusion models may yield synthetic samples that better reduce uncertainty in scarcely covered regions. Second, different transformer-based architectures and sampling techniques could provide the diversity that our current LLM struggled to deliver without overfitting. Third, a systematic study of the augmentation protocol, varying the real-synthetic sampling ratio, the number of synthetic points per acquisition, and alternative labeling schemes would clarify the optimal balance between informativeness and noise. Finally, applying this uncertainty-guided synthetic augmentation to additional real-world datasets of varying scale and complexity will test its robustness and scalability.

CRediT authorship contribution statement

Luis Moles: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Alain Andres:** Writing – review & editing, Supervision, Methodology, Conceptualization; **Goretti Echegaray:** Writing – review & editing, Supervision, Methodology; **Fernando Boto:** Writing – review & editing, Supervision, Methodology.

Data availability

Data will be made available on request.

Declaration of competing interest

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgement

The authors gratefully acknowledge the financial support given by the Basque Government (Eusko Jaurlaritza) under “Programa de apoyo a la investigación colaborativa en áreas estratégicas” (Project BISUM II: Ref. KK-2024/00048) programs. The authors would also like to acknowledge the technical support provided by all Mondragon Unibertsitatea participants in this project, providing the data used in this research.

Appendix A. Complementary Performance Metric: MAE

In this appendix, we report mean absolute error (MAE) as a complementary error-based performance metric to support the R^2 analysis presented in the main text. MAE provides an intuitive measure of average prediction error in the original scale of the target variable and serves to verify that the observed performance trends are not specific to a single metric. Across all experiments, MAE exhibits trends that are fully consistent with the R^2 results reported in Section 5.1. In particular, strategies that achieve faster uncertainty reduction and higher R^2 also yield lower MAE, while methods that struggle to reduce uncertainty show higher and more variable error levels. Importantly, the inclusion of MAE does not alter any of the conclusions drawn in the main text.

Fig. A1 shows the evolution of test-set MAE over the 20 acquisition iterations for the sampling strategies analyzed in Section 5.1. Fig. A2 reports the distribution of final MAE values across independent runs.

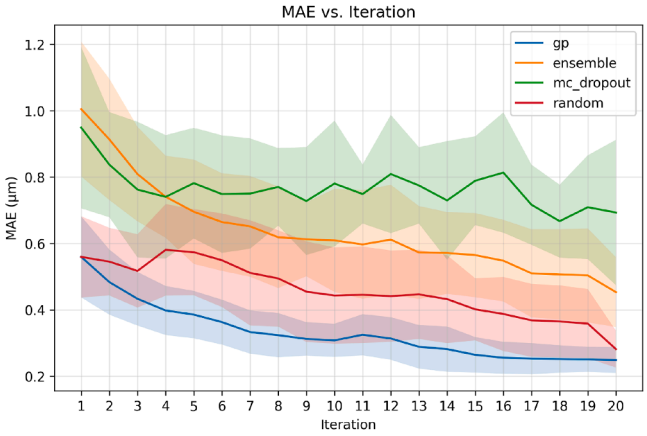


Fig. A1. Mean test-set MAE over 20 acquisition iterations for GP, XGBoost ensemble, MC-Dropout, and random sampling. Solid lines denote the mean across independent runs, and shaded bands represent the standard deviation. This figure is the MAE counterpart of Fig. 3.

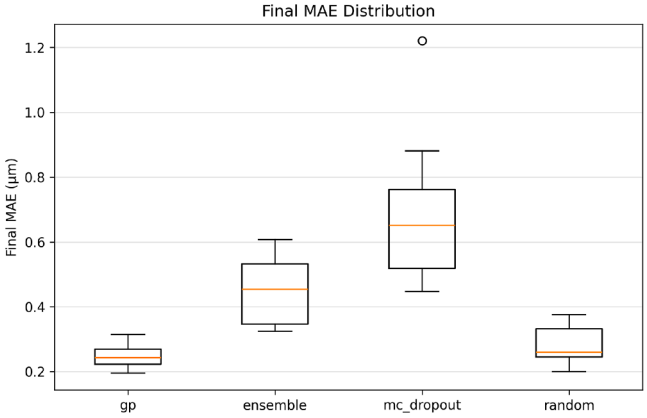


Fig. A2. Boxplots of final test-set MAE after 20 acquisition iterations, aggregated over independent runs for each sampling strategy. This figure is the MAE counterpart of Fig. 5.

Appendix B. Additional Distance To Closest Record Results

Figs. B1 and B2 report DCR histograms for the remaining random seeds. These complement the representative seed shown in Section 5.2.2 (and Fig. 7 in the main text), confirming the robustness of our conclusions. Across seeds, the qualitative trends remain consistent: the real Test→Train baseline concentrates around low distances, CTGAN exhibits broad and heavy-tailed distributions (even at higher data fractions), and VAE narrows substantially as more real data become available. LLM often shows pronounced mass at very small distances, which may indicate memorization under certain seeds.

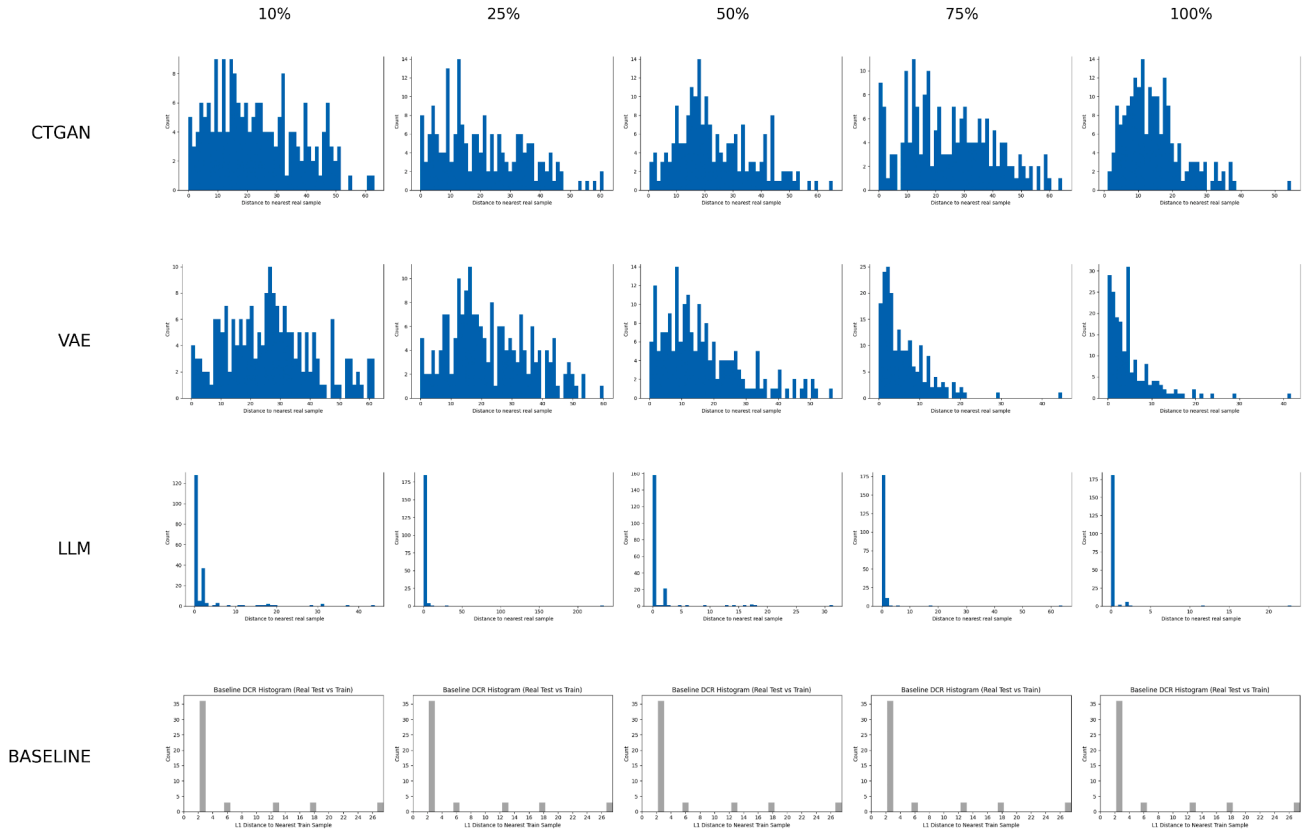


Fig. B1. DCR (L1) histograms for all generators trained with different data fractions (seed 21). Distances are computed to the nearest neighbor in the real training set. Lower values indicate synthetic points closer to genuine data in feature space.

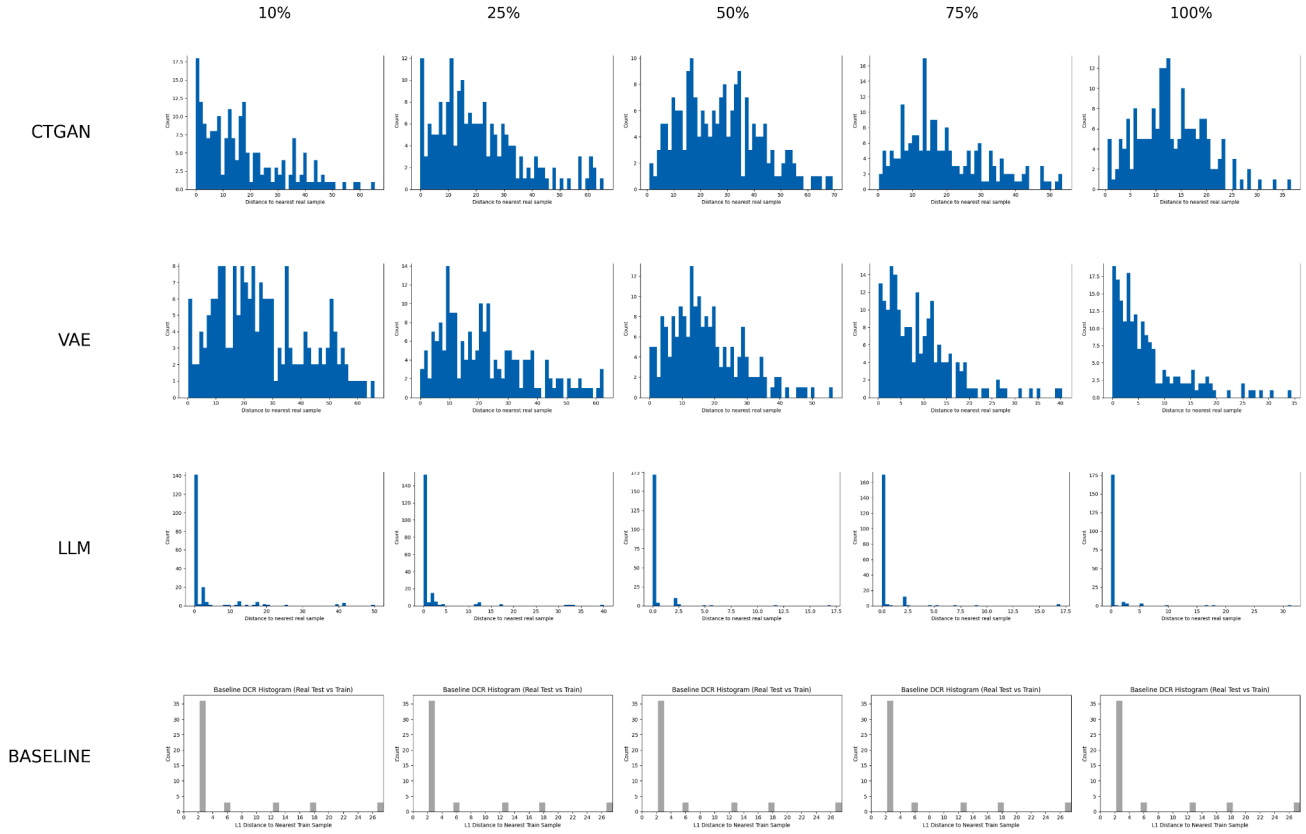
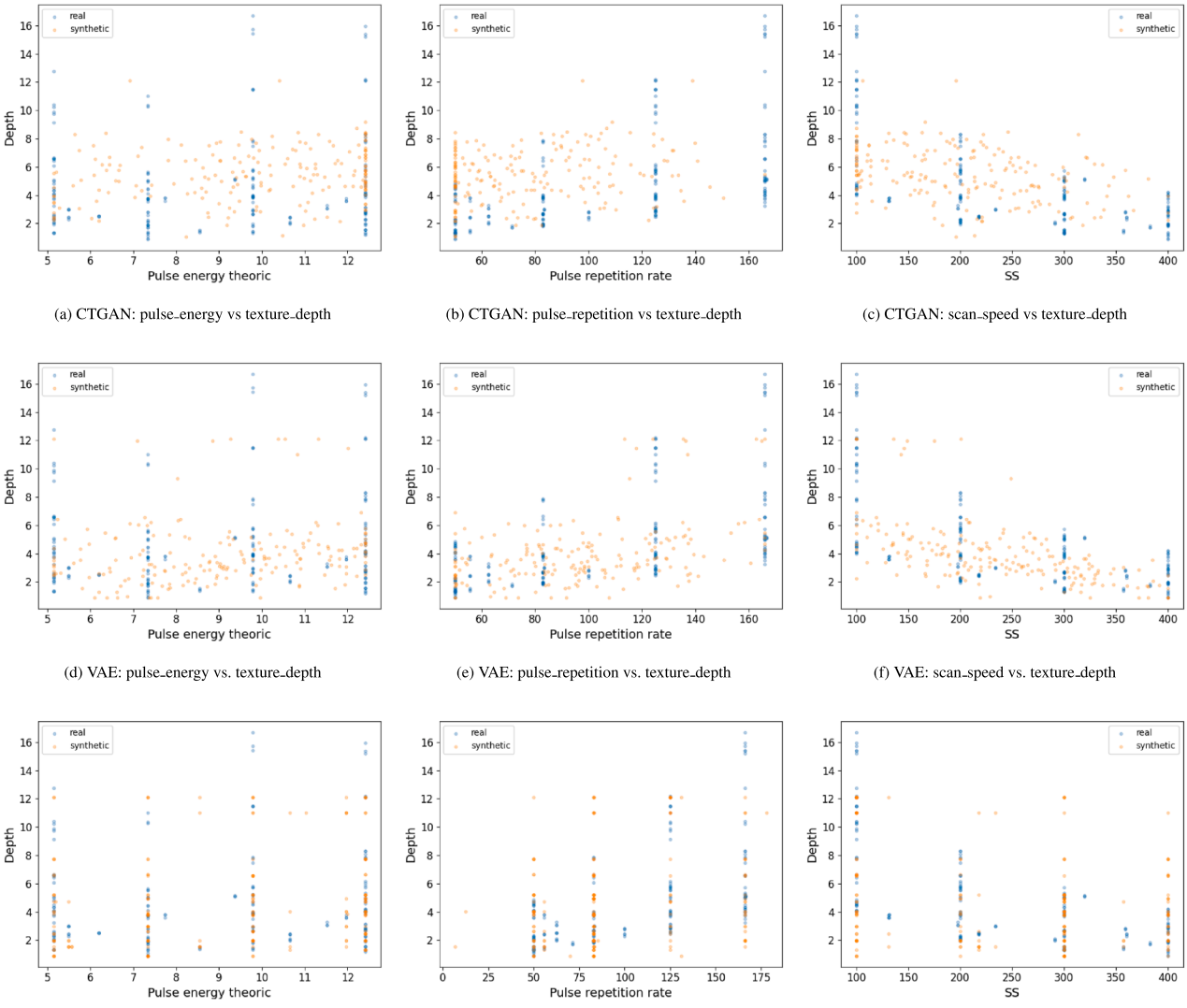


Fig. B2. DCR (L1) histograms for all generators trained with different data fractions (seed 50). Distances are computed to the nearest neighbor in the real training set. Lower values indicate synthetic points closer to genuine data in feature space.

Appendix C. Additional Bivariate Joint Plots

This appendix provides additional bivariate joint plots for data fractions other than the representative 50% case shown in Section 5.2.4. For each fraction (10%, 25%, 75%, 100%), we overlay real samples and synthetic samples generated by each model (CTGAN, VAE, LLM) to qualitatively assess how well the observed pairwise relationships are reproduced Figs. C1, C2, C3, C4.

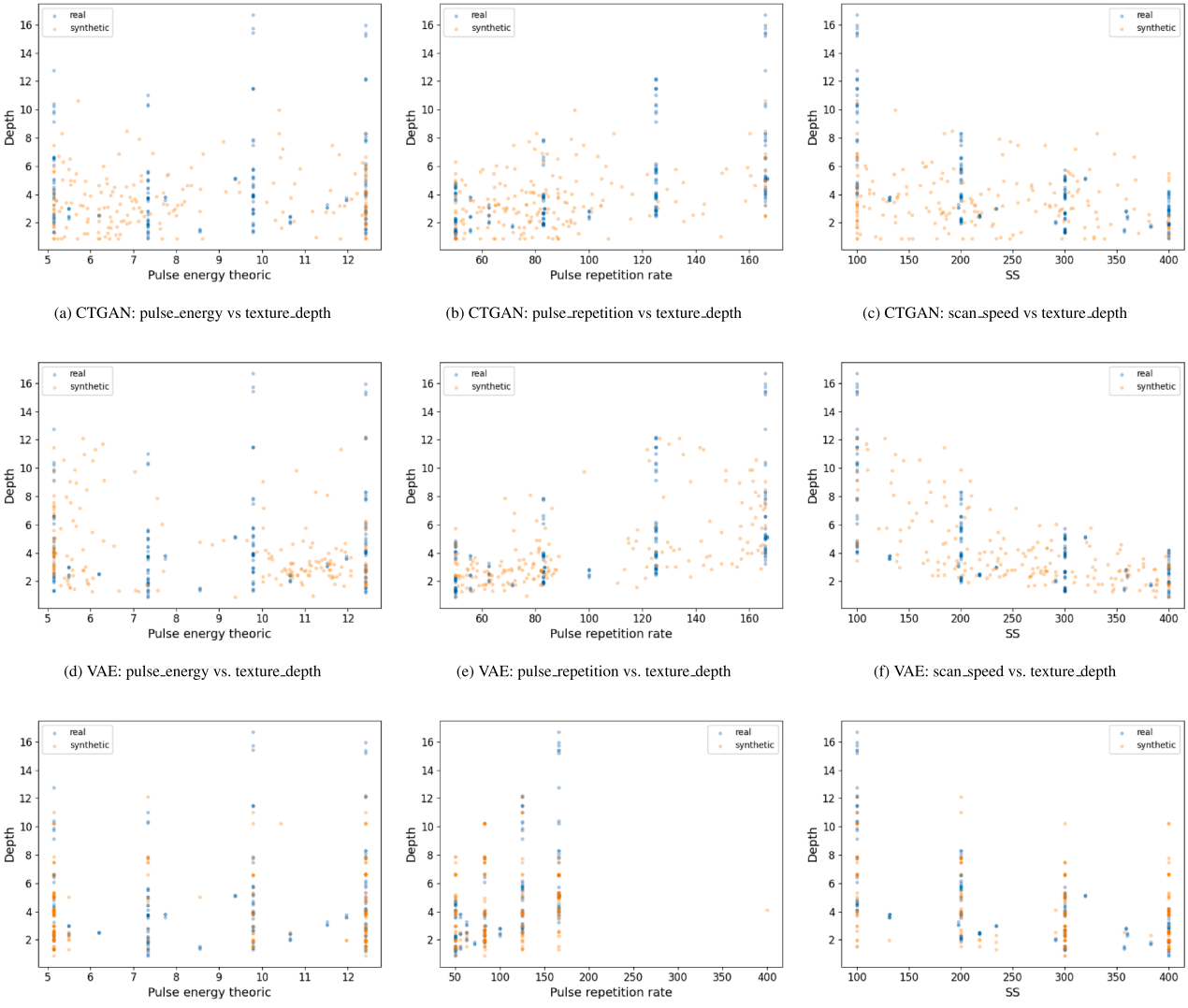


(a) CTGAN: pulse_energy vs texture_depth (b) CTGAN: pulse_repetition vs texture_depth (c) CTGAN: scan_speed vs texture_depth

(d) VAE: pulse_energy vs. texture_depth (e) VAE: pulse_repetition vs. texture_depth (f) VAE: scan_speed vs. texture_depth

(g) LLM: pulse_energy vs. texture_depth (h) LLM: pulse_repetition vs. texture_depth (i) LLM: scan_speed vs. texture_depth

Fig. C1. Bivariate joint distribution plots for the 10% data-fraction setting. Each generator (CTGAN, VAE, LLM) was trained with 10% of the real samples and then used to produce 192 synthetic points, matching the size of the full real dataset. The scatter plots overlay 192 real samples (blue) with 192 synthetic samples (orange) for the three key variable pairs: pulse energy vs. texture depth, pulse repetition rate vs. texture depth, and scan speed vs. texture depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

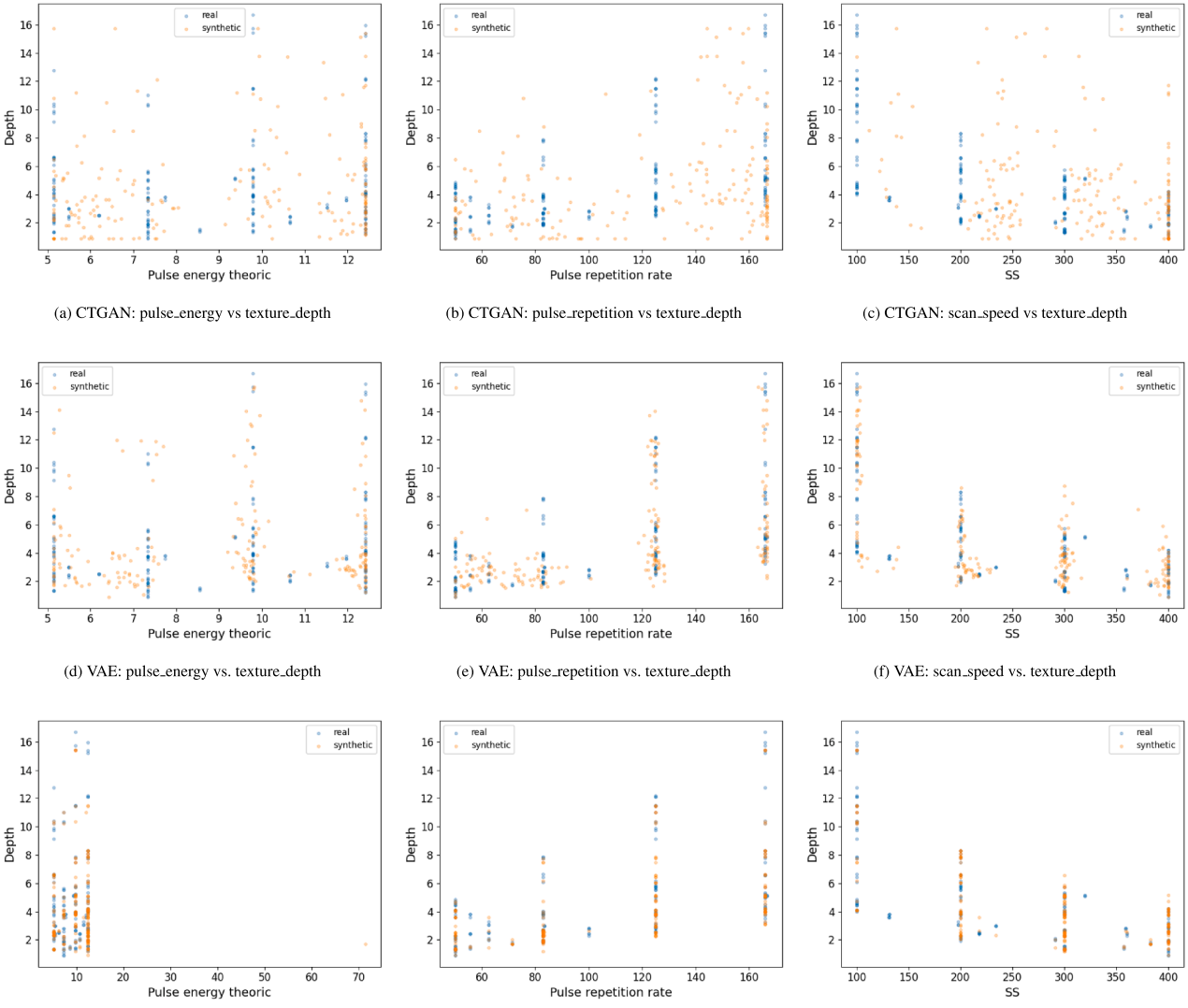


(a) CTGAN: pulse_energy vs texture_depth (b) CTGAN: pulse_repetition vs texture_depth (c) CTGAN: scan_speed vs texture_depth

(d) VAE: pulse_energy vs. texture_depth (e) VAE: pulse_repetition vs. texture_depth (f) VAE: scan_speed vs. texture_depth

(g) LLM: pulse_energy vs. texture_depth (h) LLM: pulse_repetition vs. texture_depth (i) LLM: scan_speed vs. texture_depth

Fig. C2. Bivariate joint distribution plots for the 25% data-fraction setting. Each generator (CTGAN, VAE, LLM) was trained with 25% of the real samples and then used to produce 192 synthetic points, matching the size of the full real dataset. The scatter plots overlay 192 real samples (blue) with 192 synthetic samples (orange) for the three key variable pairs: pulse energy vs. texture depth, pulse repetition rate vs. texture depth, and scan speed vs. texture depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

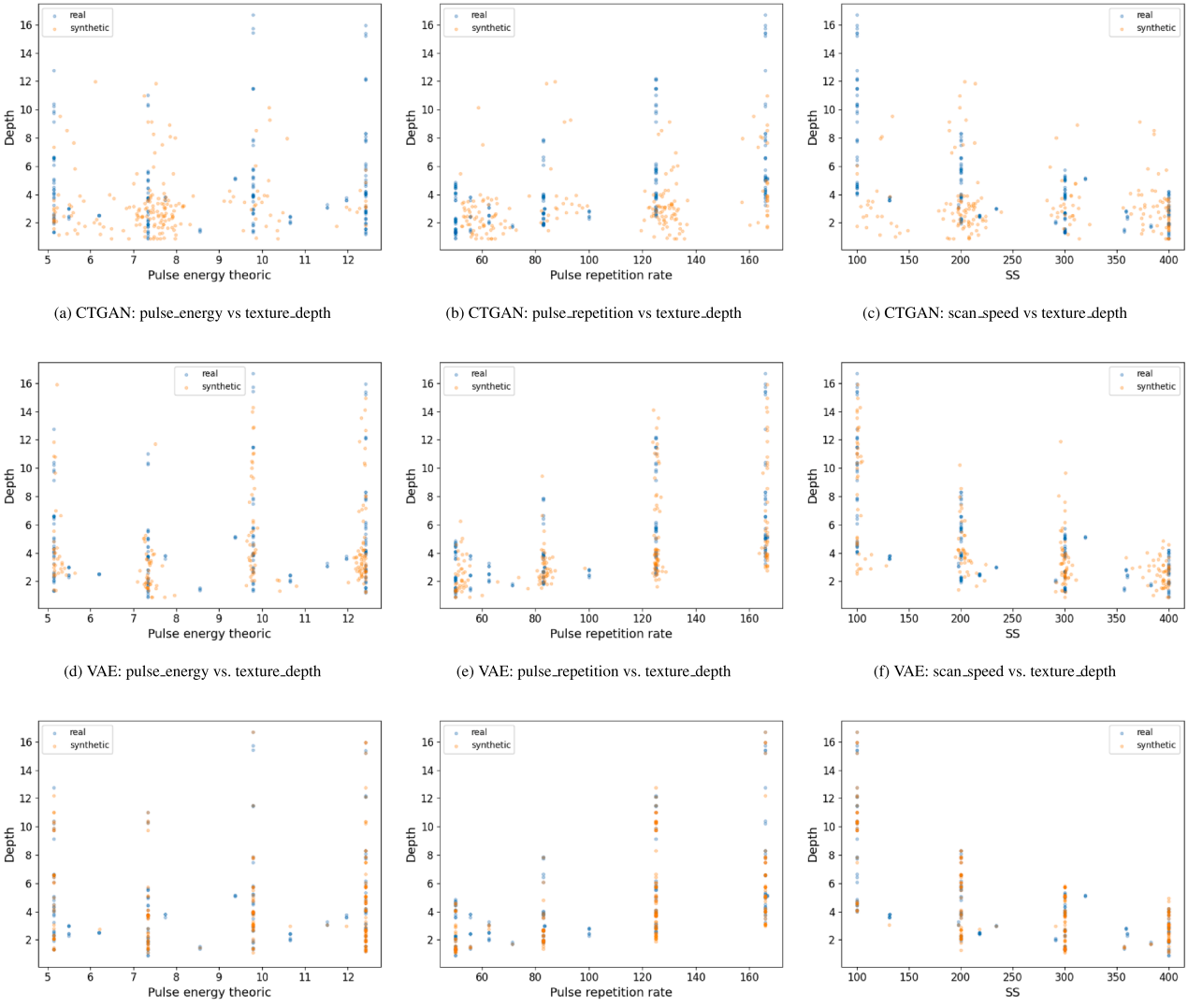


(a) CTGAN: pulse_energy vs texture_depth (b) CTGAN: pulse_repetition vs texture_depth (c) CTGAN: scan_speed vs texture_depth

(d) VAE: pulse_energy vs. texture_depth (e) VAE: pulse_repetition vs. texture_depth (f) VAE: scan_speed vs. texture_depth

(g) LLM: pulse_energy vs. texture_depth (h) LLM: pulse_repetition vs. texture_depth (i) LLM: scan_speed vs. texture_depth

Fig. C3. Bivariate joint distribution plots for the 75% data-fraction setting. Each generator (CTGAN, VAE, LLM) was trained with 75% of the real samples and then used to produce 192 synthetic points, matching the size of the full real dataset. The scatter plots overlay 192 real samples (blue) with 192 synthetic samples (orange) for the three key variable pairs: pulse energy vs. texture depth, pulse repetition rate vs. texture depth, and scan speed vs. texture depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) CTGAN: pulse_energy vs texture_depth (b) CTGAN: pulse_repetition vs texture_depth (c) CTGAN: scan_speed vs texture_depth

(d) VAE: pulse_energy vs. texture_depth (e) VAE: pulse_repetition vs. texture_depth (f) VAE: scan_speed vs. texture_depth

(g) LLM: pulse_energy vs. texture_depth (h) LLM: pulse_repetition vs. texture_depth (i) LLM: scan_speed vs. texture_depth

Fig. C4. Bivariate joint distribution plots for the 100% data-fraction setting. Each generator (CTGAN, VAE, LLM) was trained with 100% of the real samples and then used to produce 192 synthetic points, matching the size of the full real dataset. The scatter plots overlay 192 real samples (blue) with 192 synthetic samples (orange) for the three key variable pairs: pulse energy vs. texture depth, pulse repetition rate vs. texture depth, and scan speed vs. texture depth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

References

[1] B. Durakovic, Design of experiments application, concepts, examples: state of the art, Periodicals Eng. Nat. Sci. 5 (3) (2017) 421–439.
[2] F.A.C. Viana, A tutorial on latin hypercube design of experiments, Qual. Reliab. Eng. Int. 32 (5) (2016) 1975–1985.
[3] M.J. Anderson, P.J. Whitcomb, RSM simplified: optimizing processes using response surface methods for design of experiments, Productivity press, 2016.
[4] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[5] T. Voigt, M. Kohlhase, O. Nelles, Incremental DoE and modeling methodology with Gaussian process regression: an industrially applicable approach to incorporate expert knowledge, Mathematics 9 (19) (2021) 2479.
[6] B. Settles, Active learning literature survey (2009).
[7] A. Tharwat, W. Schenck, A survey on active learning: state-of-the-art, practical challenges and research directions, Mathematics 11 (4) (2023) 820.
[8] P.K. Muruganantham, S.M. Balakrishnan, Uncertainty-driven active learning in a deep semi-supervised framework for WCE image classification, Results Eng. (2025) 106174.
[9] C. Ren, Y. Xing, K.S. Patel, Application of an active learning method for cumulative fatigue damage assessment of floating wind turbine mooring lines, Results Eng. 22 (2024) 102122.
[10] C.K.I. Williams, C.E. Rasmussen, Gaussian processes for machine learning, 2, MIT press Cambridge, MA, 2006.
[11] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.
[12] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. artif. intell. res. 16 (2002) 321–357.
[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014) 2672–2680.
[14] D.P. Kingma, M. Welling, et al., An introduction to variational autoencoders, Found. Trends Mach. Learn. 12 (4) (2019) 307–392.
[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.
[16] J. Freiesleben, J. Keim, M. Grutsch, Machine learning and design of experiments: alternative approaches or complementary methodologies for quality improvement?, Qual. Reliab. Eng. Int. 36 (6) (2020) 1837–1848.
[17] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, S. Venkatesh, Bayesian optimization for adaptive experimental design: a review, IEEE Access 8 (2020) 13937–13948.
[18] B. Malluhi, R. Fezai, C. Kravaris, H. Nounou, M. Al-Rawashdeh, M. Nounou, Guided experimental design for static nonparametric modeling, Chem. Eng. Sci. 298 (2024) 120327.
[19] K. Ji, F. Chen, X. Guo, Y. Xu, J. Wang, J. Chen, Uncertainty-guided learning for improving image manipulation detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 22456–22465.
[20] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, Adv. Neural Inf. Process. Syst. 33 (2020) 12104–12114.
[21] S. Gurumurthy, R. Kiran Sarvadevabhatla, R. Venkatesh Babu, DeLiGAN: generative adversarial networks for diverse and limited data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 166–174.
[22] P.A. Apellániz, A. Jiménez, B.A. Galende, J. Parras, S. Zazo, Artificial inductive bias for synthetic tabular data generation in data-scarce scenarios, arXiv:2407.03080 (2024).
[23] J.M.D. Delgado, L. Oyedele, Deep learning with small datasets: using autoencoders to address limited datasets in construction management, Appl. Soft Comput. 112 (2021) 107836.
[24] L. Moles, A. Andres, G. Echegaray, F. Boto, Exploring data augmentation and active learning benefits in imbalanced datasets, Mathematics 12 (12) (2024) 1898.
[25] E. Mosqueira-Rey, E. Hernández-Pereira, J. Bobes-Bascarán, D. Alonso-Ríos, A. Pérez-Sánchez, Á. Fernández-Leal, V. Moret-Bonillo, Y. Vidal-Ínsua, F. Vázquez-Rivera, Addressing the data bottleneck in medical deep learning models using a human-in-the-loop machine learning approach, Neural Comput. Appl. 36 (5) (2024) 2597–2616.
[26] I. Kobyzev, S.J.D. Prince, M.A. Brubaker, Normalizing flows: an introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3964–3979.
[27] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.
[28] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. 22 (57) (2021) 1–64.
[29] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, M.-H. Yang, Diffusion models: a comprehensive survey of methods and applications, ACM Comput. Surv. 56 (4) (2023) 1–39.
[30] M. Miletic, M. Sariyar, Assessing the potentials of LLMs and GANs as state-of-the-art tabular synthetic data generation methods, in: International Conference on Privacy in Statistical Databases, Springer, 2024, pp. 374–389.
[31] X. Guo, Y. Chen, Generative ai for synthetic data generation: methods, challenges and the future, arXiv:2403.04190 (2024).
[32] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, G. Kasneci, Language models are realistic tabular data generators, arXiv:2210.06280 (2022).
[33] C. Williams, C. Rasmussen, Gaussian processes for regression, Adv. Neural Inf. Process. Syst. 8 (1995) 514–520.
[34] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.
[35] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al., XGBoost: extreme gradient boosting, R package version 0.4-2 1 (4) (2015) 1–4.
[36] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: a highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 30 (2017).
[37] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Adv. Neural Inf. Process. Syst. 32 (2019).
[38] X. Fang, W. Xu, F.A. Tan, J. Zhang, Z. Hu, Y. Qi, S. Nickleach, D. Socolinsky, S. Sengamedu, C. Faloutsos, Large language models (LLMs) on tabular data: prediction, generation, and understanding–A survey, arXiv:2402.17944 (2024).
[39] J. Bonse, S.V. Kirner, M. Griepentrog, D. Spaltmann, J. Krüger, Femtosecond laser texturing of surfaces for tribological applications, Materials 11 (5) (2018) 801.
[40] L. Orazi, G. Cuccolini, A. Fortunato, G. Tani, An automated procedure for material removal rate prediction in laser surface micromanufacturing, Int. J. Adv. Manuf. Technol. 46 (2010) 163–171.
[41] M. Benton, M.R. Hossan, P.R. Konari, S. Gamagedara, Effect of process parameters and material properties on laser micromachining of microchannels, Micromachines 10 (2) (2019) 123.
[42] S.L. Campanelli, F. Lavecchia, N. Contuzzi, G. Percoco, Analysis of shape geometry and roughness of Ti6Al4V parts fabricated by nanosecond laser ablation, Micromachines 9 (7) (2018) 324.
[43] A. Bharatish, S. Soundarapandian, Influence of femtosecond laser parameters and environment on surface texture characteristics of metals and non-metals–state of the art, Lasers Manuf. Mater. Process. 5 (2018) 143–167.
[44] Z. Wang, R. Ye, J. Xiang, The performance of textured surface in friction reducing: a review, Tribol. Int. 177 (2023) 108010.
[45] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv:1910.01108 (2019).
[46] R. Shwartz-Ziv, A. Armon, Tabular data: deep learning is not all you need, Inf. Fusion 81 (2022) 84–90.
[47] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data?, Adv. Neural Inf. Process. Syst. 35 (2022) 507–520.