

CGAN–KSR: Surface Roughness Prediction In CNC Milling From Small Samples

Thanh-Nhan Nguyen¹, Dinh-Khang Bach¹, Thai-Viet Dang², and Anh-Dung Nguyen^{1*}

¹ Department of Mechanical Engineering, School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, 10000, Vietnam

² Department of Mechatronics, School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, 10000, Vietnam

* Corresponding author. E-mail: dung.nguyennh@hust.edu.vn (A.-D. Nguyen)

Received: June 19, 2026; Accepted: August 25, 2026

Surface roughness (R_a) prediction in CNC milling is impeded by small experimental datasets that cover the cutting-parameter space sparsely and unevenly. This paper proposes a hybrid CGAN–KSR framework integrating conditional data augmentation with nonparametric smoothing under a leakage-free evaluation protocol. A Conditional GAN (CGAN) is trained on eight CNC milling experiments on Y8 carbon tool steel to learn the conditional distribution of R_a given the cutting parameters. Within each cross-validation fold the trained CGAN generates 20 condition-consistent synthetic samples at unobserved grid conditions, and a Kernel Smoothing Regression (KSR) model is fitted to the 27 combined samples. KSR is adopted not for predictive superiority but because it delivers a continuous, smooth response surface from which a process window can be read directly. Under Leave-One-Out Cross-Validation, augmentation reduces MSE from 0.363 to 0.142 (60.8%) and RMSE from 0.603 to 0.377 μm (37.4%) relative to a KSR baseline trained solely on experimental data. A percentile bootstrap 95% confidence interval for the paired MSE difference, [0.023, 0.443], is consistent with a directional improvement rather than a strongly powered or definitive statistical result, and the sign test does not reach significance at $n=8$ (6 of 8 folds, $p=0.145$). Ablation experiments show the gain is specific to conditional generation rather than sample volume, and is reproduced by augmented SVR and Gaussian Process models. Crucially, augmentation degrades uncertainty calibration: the 95% predictive intervals of an augmented Gaussian Process cover only 25% of held-out cases. The study therefore demonstrates feasibility, under the single material, tool and parameter range investigated, rather than general applicability.

Keywords: Surface roughness prediction; CNC milling; Conditional GAN (CGAN); Data augmentation; Kernel smoothing regression; Small-sample learning

© The Author(s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202701_36.014

1. Introduction

Selecting cutting conditions in advance, rather than inspecting surfaces afterwards, requires a reliable model of the process-quality relationship [1]. Surface roughness R_a is the usual target of such models [2], and a difficult one: it is governed by coupled effects of cutting parameters, process variability, material behaviour and measurement noise, giv-

ing strongly nonlinear and condition-dependent responses [3, 4]. The practical obstacle is that machining experiments are expensive, so studies typically operate with sparse and uneven coverage of the parameter space, which motivates data-efficient learning [1, 5]. One response is generative augmentation, which has been shown to enrich limited process signatures—spindle power or force traces captured during cutting—and improve downstream learning [6].

Existing roughness predictors divide by the information they consume, and the division matters for what data they need. Parameter-based regressors require no additional sensing, and range from knowledge-enhanced neural models [7] through physics-informed deep networks [8] to hybrid learning-optimization schemes [9, 10]. Signal-based predictors achieve greater robustness by consuming measured process signatures—vibration-assisted Gaussian process regression [11], cutting-force features [12]—but require an instrumented machine and, critically, a cut that has already been made. A third group addresses reliability directly, through probabilistic regression coupled to parameter selection [13], domain-incremental expansion of the input space [14], or explicit treatment of material-dependent effects [15]; deep models have also been applied to related machining processes [16, 17]. What unites the more accurate of these approaches is an appetite for data that a minimal factorial experiment cannot satisfy.

Two responses to data scarcity are established: transfer learning across related conditions [5], and enrichment of the training distribution by synthetic sample generation. Within the latter, GAN-based augmentation of process signatures improves downstream roughness prediction [6], conditional Wasserstein variants mitigate class imbalance in tool-condition monitoring [18], and virtual-sample schemes have been proposed for small-sample roughness prediction using surrogate learners [19] or interpolation [20]. Surveys caution that synthetic samples require validation, since low-fidelity samples or distribution shift can degrade the downstream estimator [21].

One qualification matters for transferring this evidence. The augmented quantity in [6] is a measured signature, which presupposes an instrumented machine and a cut already made, whereas we condition on the commanded parameters $[s, v, t]$. Generation is therefore possible before cutting, which is what makes the approach usable for planning, but the conditioning is correspondingly weaker—a power trace reflects the actual tool–workpiece interaction, a parameter triplet does not. Feasibility in the signature-based setting does not automatically carry over, and we do not treat it as doing so.

Our use of a conditional GAN rather than another generative family rests on scale, not on a claim of general superiority. Conditional variational autoencoders [22] train more stably but over-smooth the conditional distribution, which is unhelpful when the quantity of interest is precisely how R_d varies between nearby conditions; Wasserstein GANs with gradient penalty [23] require a near-optimal critic between generator updates, hard to justify with seven points; diffusion models [24] estimate a score function across many

noise levels and are correspondingly data-hungry. With a one-dimensional response and a three-dimensional condition, the adversarial objective can be carried by under a thousand parameters—the smallest capacity among these families that still models a conditional distribution rather than a conditional mean.

Nonparametric regression provides an alternative modeling paradigm for small datasets because it can adapt locally without imposing a rigid global functional form. Kernel smoothing regression (KSR) is the Nadaraya-Watson estimator, introduced independently by Nadaraya [25] and Watson [26]; these are cited here as the foundational theoretical sources for the estimator we use, not as competing methods. It estimates conditional expectations by weighting nearby samples through a kernel function and a bandwidth parameter that controls smoothness [27, 28], and has been applied to roughness modelling in end milling [29]. It carries a single scalar hyperparameter, readily selected by cross-validation—unlike LOESS, which also requires a polynomial degree, or GPR, which estimates kernel form and signal/noise variances simultaneously. Under extremely sparse coverage, however, any local estimator gives unreliable estimates in poorly sampled regions, which is what motivates augmentation.

Table 1 positions this work against the closest related studies. The studies that augment for roughness prediction either condition on measured process signatures or operate at sample sizes an order of magnitude larger than ours. More consequentially, none evaluates augmentation under a protocol that regenerates the synthetic samples inside every cross-validation fold. Where augmentation is performed once before splitting, the held-out samples have influenced the generator and reported improvements are optimistic by an unknown margin. This is not a criticism we can make from the outside: we made the same class of error in an earlier version of this work, and correcting it changed our own headline figures substantially.

Research gap and objective

Three gaps follow. (i) Generative augmentation for roughness prediction conditions either on measured signatures, unavailable at planning time, or on datasets far larger than a minimal factorial design. (ii) Where augmentation is combined with a downstream regressor, the evidence rarely separates the generator’s contribution from that of simply having more rows. (iii) Evaluation protocols seldom guarantee that generated samples carry no information about the held-out condition—precisely where small-sample studies are most vulnerable to optimism.

Objective. To determine whether conditional generative

Table 1. Positioning of the present work relative to closely related small-sample and augmentation-based roughness-prediction studies. “Leakage-free” denotes regeneration of synthetic samples inside each cross-validation fold.

Study	Conditioning input	Sample size	Augmentation	Main limitation for our setting
Knowledge-enhanced NN [7]	Cutting parameters	Moderate	None	Requires enough data to fit a deep model
GPR variants [11, 13]	Parameters + vibration	Moderate	None	Needs process sensing; no augmentation
GAN augmentation for roughness [6]	Spindle-power signature	Moderate	GAN on signatures	Requires the cut to have been made and the machine instrumented
cWGAN oversampling [18]	Condition labels	Moderate	Conditional WGAN	Classification (tool condition), not regression
Virtual sample generation [19, 20]	Cutting parameters	Small	Surrogate / interpolation	Generated values are interpolations of the fitted model
This work	Cutting parameters only	$n = 8$	CGAN, per-fold	Single material, tool and range; calibration degraded

augmentation, conditioned only on commanded cutting parameters, improves roughness prediction from a minimal $2 \times 2 \times 2$ experiment ($n = 8$); to establish under a leakage-free protocol whether any improvement is attributable to the conditional structure of the generated samples rather than to their number or to the downstream regressor; and to delimit where the resulting model may and may not be relied upon.

To pursue this objective we propose a hybrid CGAN-KSR framework. A conditional GAN learns the distribution of R_a given the exact cutting-parameter triplet $[s, v, t]$ and generates synthetic samples at unobserved grid conditions, so that the values are contextually consistent rather than drawn from the global marginal—the distinction that matters when the design is a sparse $2 \times 2 \times 2$ factorial. Kernel smoothing regression is then fitted to the augmented set. Evaluation uses a leakage-free leave-one-out protocol in which synthetic samples are regenerated from the training fold alone and the held-out sample informs neither augmentation nor model fitting.

The main contributions of this work are as follows:

- **A data-efficient framework conditioned only on planning variables:** A CGAN-KSR pipeline that generates synthetic roughness values from the commanded cutting parameters alone, so that augmentation is possible before any cut is made and without process instrumentation.
- **An evaluation protocol without spatial leakage:** Synthetic samples are regenerated inside every cross-validation fold and the grid filter is applied per fold, so that neither the response nor the location of the held-out condition informs the generator. We show this distinction is not academic: applying the filter globally, as is otherwise natural, changes the reported error by a factor

of 1.3 and reverses the outcome of the significance test.

- **Attribution of the improvement by ablation:** Interpolation- and noise-augmented controls, together with augmented variants of four competing regressors and a sweep of the augmentation ratio, establish that the gain arises from the conditional structure of the generated samples rather than from their number or from any property of the downstream regressor.
- **An explicit account of what augmentation costs:** We report that the accuracy gain is accompanied by a collapse of predictive-interval coverage from 88% to 25%, identify the mechanism, and state the resulting limits on use. To our knowledge this trade-off has not been quantified in the machining augmentation literature, and it bears directly on whether such models can inform tolerance decisions.

The smoothness of the delivered response surface, quantified in Section 3.3, is a property of kernel smoothing rather than a contribution of this work; we report it because it is the practical reason for retaining KSR, not as evidence of novelty.

Each abbreviation is defined at first use and thereafter used in abbreviated form only; a consolidated list is provided in the supplementary material.

The remainder of this paper is organized as follows. Section 2 describes the materials, experimental setup, and the proposed CGAN-KSR methodology, including implementation details and the dataset augmentation strategy. Section 3 reports evaluation results and discussion. Finally, Section 4 concludes the paper and outlines directions for future work.

2. Materials and methods

This study adopts a two-stage hybrid modeling framework that couples Conditional Generative Adversarial Networks (CGAN) for data augmentation with Kernel Smoothing Regression (KSR) for surface roughness prediction. The overall procedure consists of three steps: (i) experimental design and data acquisition, (ii) condition-consistent sample generation using CGAN, and (iii) nonparametric regression and performance evaluation using a leakage-free Leave-One-Out Cross-Validation (LOOCV) protocol.

2.1. Experimental Setup

Eight experimental data points were obtained from CNC milling trials on carbon tool steel (Y8) using a $2 \times 2 \times 2$ full-factorial design. Three cutting parameters were considered: feed rate s , spindle speed v , and depth of cut t , while the arithmetic average surface roughness R_a was measured as the response variable. The parameter levels and corresponding experimental runs are listed in Table 2. The machining trials and roughness measurements were conducted by one of the present authors and were first documented in [30], a master's thesis he supervised; that document is cited throughout as the primary record of the experimental campaign. The present work reports no new machining trials and contributes the modelling framework and its evaluation. Y8 was selected because it is one of the steels used at the collaborating tool-manufacturing workshop for punch-and-die components in sheet- and bulk-forming operations, so the parameter window studied reflects an existing production need rather than a laboratory abstraction.

All experiments were performed in the tool workshop of an industrial partner on a five-axis DECKEL MAHO DMU 105 monoBLOCK machining center with a Heidenhain iTNC 530 control, rated at 15 000 rpm maximum spindle speed and 5 μm axis positioning accuracy (Fig. 1a). Face milling was performed with a $\varnothing 25$ mm indexable shoulder mill (BAP300R 25 $\times$ 160-C25-3T) carrying $z = 3$ coated-carbide inserts of type APMT113508PDER, grade CX31NS (Fig. 1b). Workpiece blanks were $\varnothing 40 \times 70$ mm Y8 bars, rigidly clamped to ensure repeatability (Fig. 1c). Flood cooling used a 7–10% emulsion of Balaser cutting oil in water. After each milling run, surface roughness R_a was measured at three locations on the same machined surface using a Hommel-Etamic W10 contact profilometer over a 15 mm sampling length, with the stylus traversing parallel to the feed direction (Fig. 1d). Because all three traces are taken on one surface produced by a single run, their spread characterises measurement and within-surface variability, not run-to-run process variability; no run was repeated.

The instrument held a valid national metrology calibration certificate throughout the experimental campaign. The arithmetic mean of the three traces was used as the R_a value for each cutting condition, and all three individual values are reported in Table 3. Spindle-load, temperature and vibration were not instrumented, and coolant flow was not monitored; these are acknowledged as uncontrolled factors in Section 3.6.

2.2. Kernel Smoothing Regression

Kernel Smoothing Regression (KSR) is a nonparametric regression technique that estimates the conditional expectation of a response variable given input features without assuming a fixed global functional form. This local, data-adaptive property makes KSR particularly suitable for small datasets in which global parametric fits can be unstable [27, 28].

The estimated surface roughness $\hat{R}_a(x)$ for a new input vector x is computed using the Nadaraya-Watson kernel estimator [25, 26]:

$$\hat{R}_a(x) = \frac{\sum_{i=1}^n K_h(x - x_i) y_i}{\sum_{i=1}^n K_h(x - x_i)} \quad (1)$$

where x_i is the input vector of the i -th training sample, y_i is the observed surface roughness R_a , K_h is the kernel function, and h is the bandwidth parameter controlling smoothness. In this study, the Gaussian kernel was used due to its smooth characteristics and wide applicability [27, 28]:

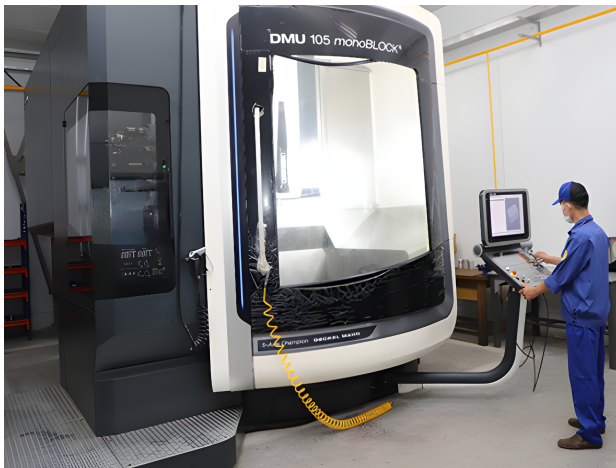
$$K_h(u) = \exp\left(-\frac{\|u\|^2}{2h^2}\right) \quad (2)$$

All input features were normalized to the range $[0, 1]$ prior to modeling. The bandwidth h , which controls the degree of smoothing, was selected by LOOCV on the training set by minimizing the MSE over the candidate grid $h \in \{0.03, 0.05, 0.07, 0.10, 0.15, 0.20, 0.30, 0.50, 0.80, 1.00\}$. The selected values are reported per fold in supplementary Table S1. The baseline selects $h = 0.15$ in six of eight folds and 0.03 in the remaining two; the augmented model selects a larger bandwidth, $h = 0.30$ in six folds and 0.20 and 0.07 in the others, and exceeds the baseline value in seven of the eight folds. The reason is explained in Section 2.7: once the interior grid conditions are populated, a bandwidth of this size averages over several nearby informative samples instead of a few distant cube vertices, so additional smoothing becomes productive rather than merely flattening.

KSR with a Gaussian kernel is preferred over related alternatives in this study for two reasons. First, compared with LOESS, KSR's single bandwidth hyperparameter is

Table 2. Cutting conditions used in the experiment. Table feed is given in mm/min with the equivalent in m/min in parentheses. Feed per tooth $f_z = s/(nz)$ and cutting speed $V_c = \pi Dn/1000$ are derived quantities for the $D = 25$ mm, $z = 3$ cutter described in Section 2.1.

No	Feed s (mm/min)	Spindle speed n (rpm)	Depth t (mm)	f_z ($\mu\text{m}/\text{tooth}$)	V_c (m/min)
1	60 (0.06)	1000	0.2	20.0	78.5
2	150 (0.15)	1000	0.2	50.0	78.5
3	60 (0.06)	1500	0.2	13.3	117.8
4	150 (0.15)	1500	0.2	33.3	117.8
5	60 (0.06)	1000	1.0	20.0	78.5
6	150 (0.15)	1000	1.0	50.0	78.5
7	60 (0.06)	1500	1.0	13.3	117.8
8	150 (0.15)	1500	1.0	33.3	117.8



(a)



(b)



(c)



(d)

Fig. 1. Experimental setup: (a) five-axis CNC milling center (DECKEL MAHO DMU 105 monoBLOCK); (b) $\varnothing 25$ mm indexable shoulder mill (BAP300R 25 $\times$ 160-C25-3T) with three coated-carbide APMT113508PDER inserts; (c) workpiece sample (Y8 carbon tool steel); (d) surface roughness measurement device (Hommel-Etamic W10 contact profilometer).

Photographs of the authors' own experimental campaign, previously published in [30].

more tractable for LOOCV-based selection when n is very small (LOESS requires tuning both a span and polynomial degree). Second, compared with GPR, KSR does not require simultaneous optimization of kernel-form, signal-variance,

and noise-variance hyperparameters—fitting three GPR hyperparameters on $n = 7$ –8 samples is prone to overfitting or degenerate solutions. KSR's single bandwidth provides a more robust model-selection criterion under extreme data

scarcity.

2.3. Conditional GAN for Data Augmentation

Conditional Generative Adversarial Networks (CGANs) extend GANs by conditioning the generation process on observed inputs [31, 32]. In this study, a CGAN is used to model the conditional distribution of surface roughness (R_a) given cutting parameters (feed rate s , spindle speed v , and depth of cut t) and to generate condition-consistent synthetic R_a samples for data augmentation.

The CGAN is trained by solving the conditional mini-max game:

$$\min_G \max_D \mathbb{E}_{y \sim p_{data}(y|x)} [\log D(y, x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z, x), x))] \quad (3)$$

The generator $G(z, x)$ is a fully connected feedforward network that takes as input the concatenation of a noise vector z and the conditioning vector $x = [s, v, t]$, and outputs a synthetic R_a value. The discriminator $D(R_a, x)$ receives the pair (R_a, x) and outputs the probability that the R_a value is real rather than generated.

Model performance was assessed using a leakage-free Leave-One-Out Cross-Validation (LOOCV) protocol on the experimental measurements ($n = 8$). For each fold, one experimental sample was held out as the test case, while the remaining seven samples formed the training set. Within each fold, the CGAN was trained only on the seven training samples and then used to generate synthetic samples on the predefined grid. The grid filter is applied dynamically within each fold: only the seven active training coordinates are removed from the 27-point candidate grid, leaving 20 generation points. One of these coincides with the held-out cutting condition, for which the CGAN therefore produces a synthetic sample without ever having observed its measured R_a . Excluding the held-out coordinate a priori—as a global, pre-loop filter would do—would disclose the location of the test condition to the generator and constitutes a subtle form of spatial information leakage; the present protocol avoids it. A KSR model was then trained on the augmented training set (7 real + 20 synthetic), with the bandwidth selected by inner LOOCV on the training set. The trained KSR model was used to predict R_a for the held-out experimental sample, and the prediction error was recorded. For comparison, a baseline KSR model was trained on the same seven experimental samples without augmentation.

What this evaluation setting does and does not test. Because the held-out coordinate's input x remains in the generation grid, the framework is evaluated at a known

input condition with an unobserved response, not at a completely unseen input condition: the generator conditions on x , but the corresponding measured R_a is withheld from both the generator and the KSR fit throughout. This does not constitute conventional target leakage, but it is a more specific claim than unseen condition or generalisation can suggest, and we avoid those terms for this evaluation elsewhere in the paper—the leave-one-material-out protocol proposed as future work in Section 4 is the evaluation that would test genuinely unseen input conditions. The practical motivation for retaining the coordinate is that a process engineer querying the model has already chosen x : the cutting parameters are a decision, not an unknown, so a deployment-realistic evaluation should let the generator condition on the queried point exactly as it would at inference time; only the resulting R_a is genuinely unknown, to the engineer and to the model alike. Because a synthetic sample generated at the held-out coordinate can materially influence the subsequent KSR prediction, Section 3.4 decomposes the reported accuracy into the generator's own contribution and the smoothing step's, so that the LOOCV performance figures below are not over-read as evidence of prediction at a structurally unconditioned input.

Overall performance was summarized across the eight test folds using MSE and MAE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{R}_{a,i} - R_{a,i})^2 \quad (4)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{R}_{a,i} - R_{a,i}| \quad (5)$$

To avoid information leakage, the evaluation procedure is summarized in Algorithm 1 and Fig. 2.

By conditioning the generation process on cutting parameters, the CGAN produces contextually relevant synthetic R_a values and improves the coverage of the cutting-parameter space, which supports more stable response-surface modeling under small-sample conditions [6] (see Fig. 3).

2.4. Baseline KSR Model on the Original Dataset

The following subsections detail the implementation of the proposed CGAN-KSR framework, covering (i) the baseline KSR model trained on the experimental dataset, (ii) the CGAN architecture and training settings, (iii) the full-factorial candidate grid used for synthetic-sample generation, and (iv) the use of the augmented dataset for response-surface visualization.

A baseline KSR model was first fitted using the eight experimental samples obtained from the $2 \times 2 \times 2$ full-factorial milling trials. Each sample corresponds to a

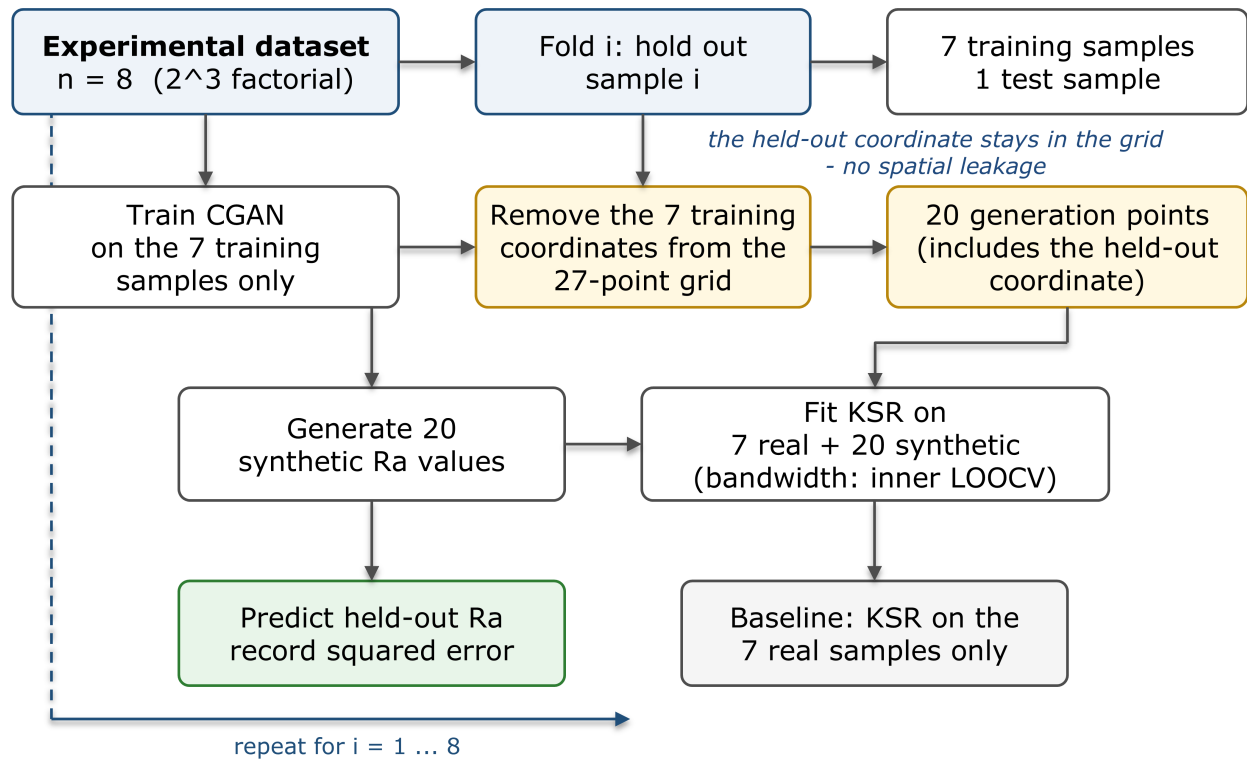


Fig. 2. Workflow of the leakage-free Leave-One-Out Cross-Validation (LOOCV) procedure used to evaluate KSR prediction accuracy on experimental measurements ($n = 8$) under both the baseline (real-only training) and CGAN-augmented training settings.

Algorithm 1. Leakage-free LOOCV evaluation protocol.

Require: Experimental dataset D_{real} ($n = 8$) and candidate cutting-condition grid G

Ensure: MSE and MAE computed on the 8 held-out experimental samples

- 1: **for** $i = 1$ to 8 **do**
- 2: Hold out the i -th experimental sample as test; use the remaining 7 samples as training set.
- 3: Train CGAN on the 7 training samples only.
- 4: Form the per-fold generation set $G_i \leftarrow G \setminus \{\text{the 7 training coordinates}\}$, so $|G_i| = 20$.
 ▷ *dynamic, per-fold filtering; the held-out coordinate remains in G_i*
- 5: Generate one synthetic sample at each condition in G_i using the trained CGAN.
- 6: Select KSR bandwidth by inner LOOCV on the augmented training set.
- 7: Train KSR on the augmented training set (7 real + 20 synthetic).
- 8: Predict R_a for the held-out experimental sample; record the prediction error.
- 9: **Baseline:** repeat the loop with no augmentation (KSR trained on the 7 real samples only).

unique combination of feed rate s , spindle speed v , and depth of cut t , with R_a measured as the response. The experimental dataset (Table 3) contains the three input parameters (s, v, t) and the corresponding R_a values (average of three measurements). Using these eight samples, KSR was used to visualize the roughness response surfaces under three representative fixed-parameter conditions (Fig. 4).

As expected with sparse sampling, the baseline KSR surfaces can be locally unstable and exhibit sharp variations in poorly supported regions, which motivates enriching the parameter space with condition-consistent synthetic samples.

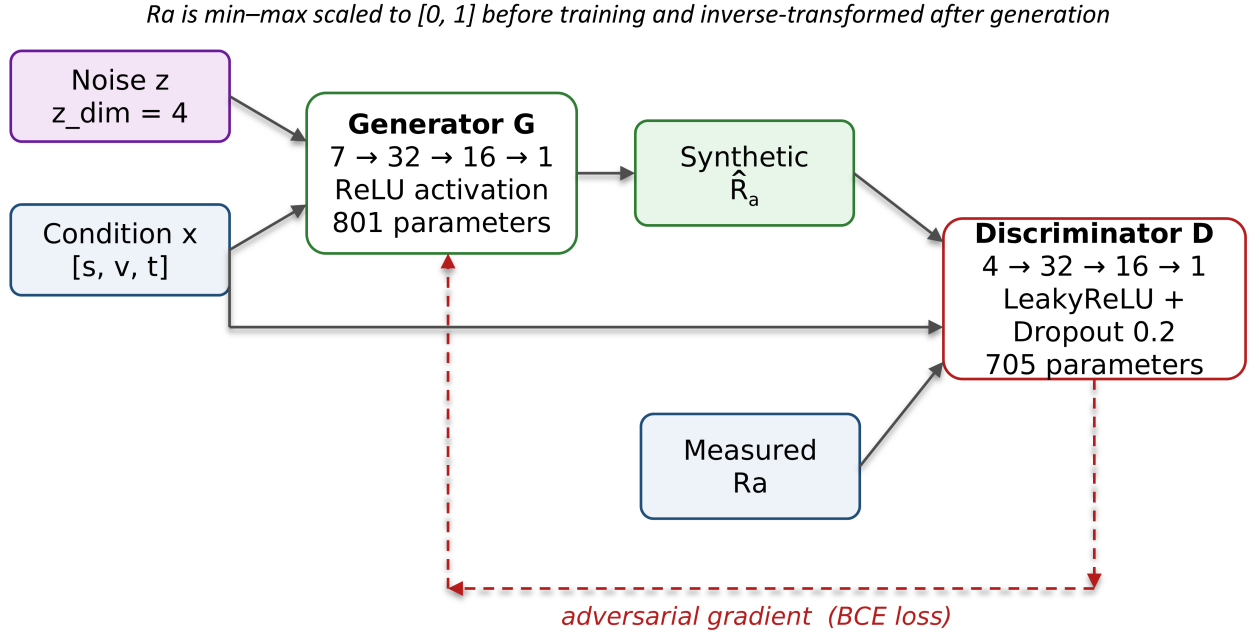


Fig. 3. Architecture and training workflow of the CGAN model for surface roughness prediction. The generator produces R_a values from noise and cutting parameters, while the discriminator distinguishes between real and generated outputs. The conditioning vector $x = [s, v, t]$ is supplied to both networks: to the generator directly, and to the discriminator via the lower path shown, consistent with the conditional discriminator $D(R_a, x)$ used throughout this paper.

Table 3. Original machining data. R_{a1} , R_{a2} and R_{a3} are three roughness measurements obtained at different locations on the same machined surface, not independent experimental replicates; $R_{a\text{ avg}}$ is their arithmetic mean.

No.	s (mm/min)	v (rpm)	t (mm)	R_{a1} (μm)	R_{a2} (μm)	R_{a3} (μm)	$R_{a\text{ avg}}$ (μm)
1	60	1000	0.2	0.242	0.329	0.152	0.241
2	150	1000	0.2	0.264	0.188	0.136	0.196
3	60	1500	0.2	1.777	2.309	1.282	1.789
4	150	1500	0.2	0.356	0.590	0.679	0.542
5	60	1000	1.0	0.777	0.945	0.549	0.757
6	150	1000	1.0	0.743	0.836	0.899	0.826
7	60	1500	1.0	2.032	0.974	1.569	1.525
8	150	1500	1.0	0.636	0.886	0.776	0.766

2.5. CGAN Architecture and Training Configuration

A Conditional Generative Adversarial Network (CGAN) was implemented in PyTorch to generate synthetic R_a values conditioned on machining parameters. Both the generator and discriminator were modeled as fully connected networks with two hidden layers, using ReLU (generator) and LeakyReLU (discriminator) activations; dropout with rate 0.2 was applied after each discriminator hidden layer to reduce overfitting. The generator has 801 trainable parameters and the discriminator 705; this deliberately small capacity matches the extreme data constraint. Training used Adam with a learning rate of 0.001, full-batch updates, and a fixed budget of 3000 epochs in every ex-

periment reported in this paper. The random seed was fixed at $42 + i$ for LOOCV fold i and at 42 for full-dataset refits, and identical hyperparameters were used in all folds. This per-fold offset exists only to give each fold a distinct, reproducible seed; it is not a stability assessment, since it does not repeat training within a single fold. That question is addressed directly by an independent-retraining analysis reported below (Section 3.6) and in the supplementary material.

Both the conditioning features and the response were min-max scaled to $[0, 1]$ before training, using statistics computed from the training fold only; generated values were inverse-transformed to μm and clipped at zero, since

Fig. 4 KSR response surfaces — trained on 8 experimental samples

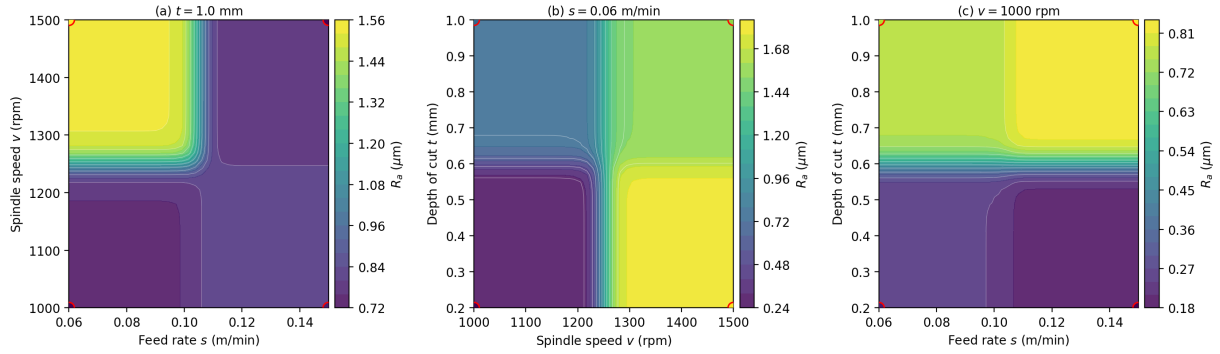


Fig. 4. Predicted R_a response surfaces from KSR trained on 8 experimental samples. Each subplot shows variation with two parameters while one is fixed: (a) $t = 1.0$ mm; (b) $s = 60$ mm/min; (c) $v = 1000$ rpm. Red circles indicate experimental data points projected onto each plane. Note: Surfaces are for visualization only; performance is reported via leakage-free LOOCV (Table 5).

R_a cannot be negative.

Loss functions. Writing $D(R_a, x) \in [0, 1]$ for the discriminator's estimated probability that R_a is a real (not generated) measurement under condition $x = [s, v, t]$, and $G(z, x)$ for the generator's output given noise $z \sim N(0, I)$ and condition x , each training iteration performs one Adam step on each of

$$L_D = \frac{1}{2} \left[BCE(D(R_a^{\text{real}}, x), 1) + BCE(D(G(z, x), x), 0) \right],$$

$$L_G = BCE(D(G(z, x), x), 1), \quad (6)$$

with independently sampled z for the D and G updates and BCE the binary cross-entropy. L_G is the non-saturating generator objective— G maximises $\log D(G(z, x))$ rather than minimising $\log(1 - D(G(z, x)))$ as in the original minimax formulation—adopted because it supplies stronger gradients while the generator is still weak. It shares the fixed point $D \equiv 0.5$ with the minimax game, but its convergence to that point is not covered by the same game-theoretic guarantee. Accordingly, we refer to $D_{\text{loss}} \rightarrow -\ln 0.5$ throughout this paper as an empirical convergence criterion, not as a proven Nash equilibrium.

Rather than stopping on visual inspection of the loss curves, which is not reproducible, we fix the epoch budget in advance and verify convergence afterwards by this criterion, applied identically to every fold: the mean discriminator loss over the final 1000 epochs must lie within 0.02 of $-\ln 0.5 = 0.6931$. All eight folds satisfy it (Section 3.4). A fixed budget with post-hoc verification is preferred to early stopping, which would consume observations that cannot be spared at $n = 7$.

The generator input concatenates a noise vector of dimension $z_{\text{dim}} = 4$ with the condition $x = [s, v, t]$. This value was fixed a priori as an architectural default, before any evaluation; it was not chosen from the sensitivity results in Table 4, which were produced later and are therefore reported as a robustness check rather than a selection procedure. Performance varies by less than a factor of 1.4 across the range examined, so no conclusion rests on the value chosen. Readers of the previous version should note that the sensitivity figures reported there came from the earlier, defective protocol at a different epoch budget; under the corrected protocol the ordering changes and $z_{\text{dim}} = 4$ happens to give both the lowest error and the smallest conditional dispersion. We place no weight on this—the choice preceded these numbers, and we would have reported the same configuration had it ranked worst—and we deliberately do not re-tune on these folds, since selecting a hyperparameter on the folds that provide the headline estimate would be selection on the evaluation criterion. The rationale for preferring this compact conditional architecture over larger generative families—WGAN-GP [23], conditional VAEs [22] and diffusion models [24]—is given in the Introduction and rests on the capacity that seven training points can support, not on a comparison performed here. Measured training and inference costs are reported in Section 3.7.

Noise-dimension robustness check (z_{dim} fixed a priori at 4):

- $z_{\text{dim}} = 2$: LOOCV-MSE = 0.1670, RMSE = 0.409 μm , $\sigma_{\text{cgan}} = 0.036 \mu\text{m}$
- $z_{\text{dim}} = 4$ (used): LOOCV-MSE = 0.1424, RMSE = 0.377 μm , $\sigma_{\text{cgan}} = 0.030 \mu\text{m}$

Table 4. CGAN architecture, training configuration, and noise-dimension sensitivity. σ_{cgan} = average conditional standard deviation from $M = 10$ independent realizations per condition. For scale, the sample standard deviation of the eight measured R_a values is $0.566 \mu\text{m}$.

Component / Parameter	Detail
Framework	PyTorch
Input features	Feed rate s , spindle speed v , depth of cut t
Output	Surface roughness R_a (average)
Noise vector z	Dimension $z_{dim} = 4$
Generator	3 layers: [Input(7) $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ 1], ReLU
Discriminator	3 layers: [Input(4) $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ 1], LeakyReLU + Sigmoid
Discriminator dropout	0.2 after each hidden layer
Loss function	Binary cross-entropy (BCE)
Optimizer	Adam (lr 0.001), full batch
Epochs	3000 (fixed budget, convergence verified post hoc)
Random seed	42 + i for LOOCV fold i ; 42 for full-dataset refit
Trainable parameters	Generator 801; discriminator 705
Synthetic data	20 per fold (27-point grid less the 7 training coordinates)

- $z_{dim} = 8$: LOOCV-MSE = 0.1957, RMSE = $0.442 \mu\text{m}$, $\sigma_{cgan} = 0.038 \mu\text{m}$

2.6. Data Augmentation Strategy

Because the experimental dataset contains only eight samples, a systematic augmentation strategy was adopted to improve coverage of the cutting-parameter space. Rather than random sampling, candidate conditions were selected using a structured full-factorial grid within the observed parameter ranges. Specifically, three discrete levels were chosen for each parameter to form a $3 \times 3 \times 3$ grid, including intermediate levels that were not present in the original experiments:

- Feed rate: 60, 90, 150 mm/min (0.06, 0.09, 0.15 m/min)
- Spindle speed: 1000, 1200, 1500 rpm
- Depth of cut: 0.2, 0.6, 1.0 mm

Combining all levels yields a $3 \times 3 \times 3 = 27$ -point candidate grid, to which two distinct filters are applied. *Per fold (evaluation)*: only the seven training coordinates are removed, leaving 20 generation points; all reported performance figures derive from this setting. *Full dataset (visualisation)*: all eight experimental conditions are removed, leaving 19 points, used only to draw Fig. 5 and Table S2 and never to compute a metric. The intermediate levels ($s = 90 \text{ mm/min}$, $v = 1200 \text{ rpm}$, $t = 0.6 \text{ mm}$) are interior points chosen to improve space-filling coverage; apart from $t = 0.6 \text{ mm}$ they are not arithmetic midpoints of the factorial levels, and no property of the method depends on their being so.

The combined dataset used for response-surface visualization—the 8 experimental samples together with 19 CGAN-generated samples on the grid—is tabulated in full as supplementary Table S2.

2.7. Why Conditional Augmentation Helps a Local Estimator—and When It Fails

The mechanism is a bias-variance trade-off. The Nadaraya-Watson estimate at a query point x_0 is a locally weighted average whose bias grows with the bandwidth h and whose variance falls with the effective local sample size $n_{eff}(x_0) = \sum_i K_h(x_i - x_0)$. With seven observations at the vertices of the normalised cube, every held-out condition lies at unit distance or more from all of them, and no bandwidth avoids both problems: one small enough to keep bias low leaves n_{eff} near unity, so the estimate collapses onto the few adjacent vertices, and one large enough to stabilise the variance pulls it toward the global mean. Either way the baseline cannot localise, which is why its predictions are near-constant: across the eight folds they span only $0.573\text{--}0.964 \mu\text{m}$ (s.d. 0.132) against a measured span of $0.196\text{--}1.789 \mu\text{m}$ (s.d. 0.566).

Populating the interior grid conditions raises n_{eff} at and around the query point. The consequence for bandwidth selection runs opposite to the sparse case and is worth stating explicitly, because it is easily predicted the wrong way round: the augmented model selects a *larger* bandwidth than the baseline, not a smaller one (median 0.30 against 0.15; larger in seven of eight folds, supplementary Table S1). A bandwidth of that size is no longer a pull toward the global mean, because what it now averages over is a set of nearby grid points carrying conditional information rather than a handful of distant vertices. The extra smoothing damps the generator’s per-point error while preserving the local trend, which is also why the augmented prediction tracks the generator’s value at the held-out coordinate closely without being identical to it (Section 3.4).

The same argument states when the mechanism must fail. Added samples reduce variance only if their responses are conditionally informative; where the generator’s condi-

tional mean is biased, the augmented estimator inherits that bias while its apparent precision increases, since it cannot distinguish generated samples from measurements. Three consequences follow, all of which we observe: the benefit vanishes when the added responses carry no information beyond the baseline's own fit (interpolation ablation, Section 3.1), reverses when they are uninformative about the condition (noise ablation), and uncertainty estimates from the augmented sample are over-confident (Section 3.5). A fuller derivation is given in the supplementary material.

2.8. Retraining the KSR Model with Augmented Data

Using the 19 CGAN-generated samples, KSR was refit on the combined 27-sample dataset (8 real + 19 synthetic) to obtain smoother response surfaces for visualization. The same normalization and bandwidth-selection procedure described in Section 2.2 was applied. This full-dataset refit is intended to illustrate how increased coverage stabilizes the estimated response surface across the parameter space; it is not used for performance reporting. The resulting response surfaces are shown in Fig. 5 under three representative fixed-parameter conditions, enabling direct qualitative comparison with Fig. 4.

3. Results and discussion

Two distinct models are reported in this paper, and the distinction governs how every number below should be read. The *evaluated model* is the per-fold model defined by Algorithm 1: it is rebuilt from scratch in each of the eight folds, never sees the held-out measurement, and is the sole basis for every performance figure we report. The *delivered model* is the full-dataset refit on all eight experimental and 19 generated samples; it underlies the response surfaces of Fig. 5 and supplementary Table S2, serves only as a representative visualisation of surface smoothness, and carries no cross-validated error estimate. No performance claim in this paper is derived from the delivered model.

3.1. Evaluation Protocol, Baselines and Ablations

The $2 \times 2 \times 2$ factorial design ($n = 8$) represents a minimally instrumented setting, and the results are lower-bound feasibility evidence rather than general-purpose performance claims. To avoid the data-volume asymmetry that would confound a comparison between a 27-sample model and 7-sample baselines, every competing regressor is evaluated in both regimes: trained on the seven real samples alone, and trained on the same augmented set of 7 real + 20 synthetic samples supplied to the proposed method. The held-out experimental sample is never used

in augmentation, bandwidth or hyperparameter selection, or model fitting.

Two ablations isolate the mechanism. In *interpolation augmentation*, the 20 synthetic responses are taken from the KSR baseline's own predictions at the same grid coordinates, so the augmented set gains coordinates but no information the baseline did not already possess. In *noise augmentation*, responses are drawn from $N(\hat{y}_{tr}, s_{tr}^2)$ independently of the condition, retaining volume but discarding conditional structure; because this depends on a random draw, it is repeated 20 times and reported as mean $\pm$ standard deviation.

The bootstrap baseline is specified as follows. Resampling is performed strictly within each LOOCV fold: from the seven training observations of that fold, seven observations are drawn with replacement, $B = 50$ times. KSR is refitted on every bootstrap sample, each with its own bandwidth selected by inner cross-validation on that resample, and the 50 predictions at the held-out point are averaged. No new cutting conditions are introduced at any stage—the resampled sets contain only coordinates already present in the training fold. This is precisely why the bootstrap baseline serves as a control for sample count without design enrichment, and why its failure to approach the augmented models (Table 5) is informative.

3.2. Prediction Accuracy under Leakage-Free LOOCV

Table 5 summarizes the leakage-free LOOCV prediction errors. Relative to the real-only KSR baseline, CGAN augmentation reduces MSE from 0.3632 to 0.1424 (60.8%), RMSE from 0.603 to 0.377 μm (37.4%), and MAE from 0.5182 to 0.3180 (38.6%). The improvement is not confined to KSR: augmenting SVR and Gaussian Process Regression with the same synthetic samples yields 0.1473 and 0.1531 respectively, both statistically indistinguishable from the proposed model. Five regressors are evaluated in both regimes (KSR, SVR, GPR, Random Forest and linear regression); on real data alone they span 0.2658–0.3771, after augmentation 0.1424–0.2350, and every one of the five improves, by between 38% and 61%. We therefore attribute the improvement to the augmentation, not to the choice of downstream regressor, and we do not claim that CGAN-KSR is the best-performing method: the 0.0049 gap to augmented SVR is far smaller than the resolution of an $n = 8$ comparison. A conventional response-surface model attains a still lower MSE (0.1404, against our 0.1424); we do not treat that comparison as evidence of CGAN-KSR's superiority either, for the structural reasons given below ("The saturated RSM model", below), and the framework's contribution is better stated as conditional data augmentation and response-

Fig. 5 KSR response surfaces — trained on 27 samples (8 real + 19 CGAN-generated)

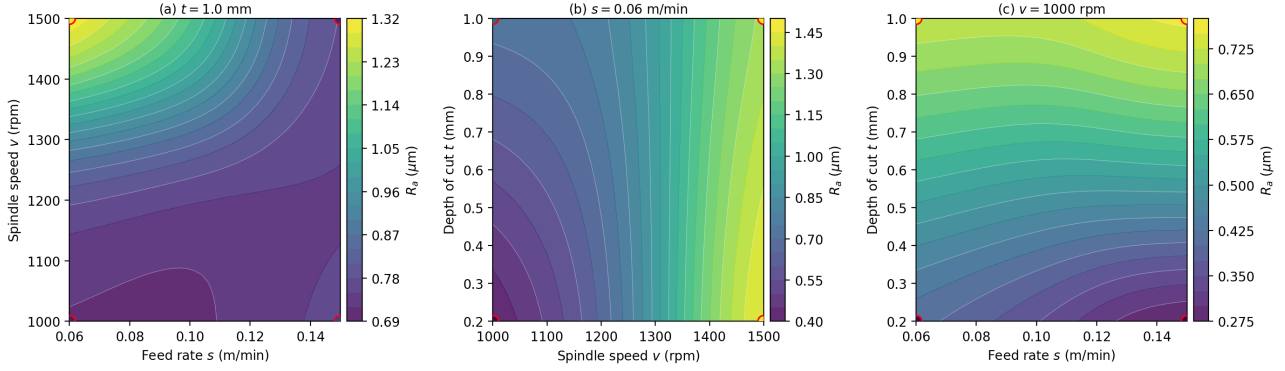


Fig. 5. Predicted R_a response surfaces from KSR trained on 27 samples (8 experimental + 19 CGAN-generated). Each subplot shows variation with two parameters while one is fixed: (a) $t = 1.0 \text{ mm}$; (b) $s = 60 \text{ mm/min}$; (c) $v = 1000 \text{ rpm}$. Red circles indicate experimental data points projected onto each plane. Compared with Fig. 4, the augmented surfaces exhibit reduced local variance and smoother transitions across the parameter space. Note: Surfaces are for visualization only; performance is reported via leakage-free LOOCV (Table 5).

surface construction under small-sample conditions than as outright predictive superiority over every alternative in Table 5.

Two results run counter to expectation. First, Gaussian Process Regression on the seven real samples (0.3521) is not a stronger small-sample baseline than KSR (0.3632); the two are indistinguishable. Second, the ablations show the gain is specific to conditional generation: augmenting with the baseline’s own interpolated values leaves performance essentially unchanged (0.3562 versus 0.3632; paired ΔMSE 0.0070, 95% CI $[-0.007, 0.021]$), so adding samples at new coordinates achieves nothing unless their responses carry information the baseline lacks, while unconditional noise is actively harmful (0.5816 ± 0.199). The contrast between CGAN-KSR and the interpolation ablation is itself significant (ΔMSE 0.2138, 95% CI $[0.022, 0.427]$).

The saturated RSM model, and why its error is not a measure of skill. Response-surface models are the conventional tool in this setting, so we include two. The power-law form $R_a = C s^{b_1} v^{b_2} t^{b_3}$, fitted by linear regression on logarithms, is the model used in the source study [30] and is standard in the machining literature; it gives MSE 0.4483, worse than the KSR baseline. Reassuringly, our per-fold refits recover the published exponents exactly: averaged over the eight folds we obtain $b_1 = -0.55$, $b_2 = +2.25$, $b_3 = +0.43$, against -0.55 , $+2.25$, $+0.43$ reported in [30]. This is an independent check that our handling of the primary data reproduces the original analysis.

The first-order-with-interactions model requires more care, because it attains the lowest MSE in Table 5 (0.1404)

and we do not wish that number to be read as a predictive result. On a 2^3 design this model has seven parameters and each training fold has seven points, so it is saturated: it interpolates the training fold exactly, with zero residual degrees of freedom. The consequence is visible in the errors themselves. Every one of the eight leave-one-out residuals has identical magnitude, $0.3747 \mu\text{m}$, with a standard deviation across folds of $< 10^{-6}$. That is not a coincidence: on this design the only term omitted from the model is the three-factor interaction, and the leave-one-out error is exactly the omitted three-factor contrast, $8|b_{123}| = 0.3747 \mu\text{m}$. The model’s cross-validated error therefore measures a single fixed quantity—one interaction effect—rather than any ability to predict.

Two further observations confirm that it cannot serve the purpose of this work. It predicts a negative roughness at one held-out condition ($-0.179 \mu\text{m}$, fold 2), which is physically impossible. And having no residual degrees of freedom, it provides no basis for estimating uncertainty and no validation whatsoever at the interior conditions where a process window must actually be read. We report its MSE because it is the lower figure and withholding it would be selective, but a saturated design model is not a competitor to a method intended to interpolate the parameter space.

Statistical assessment. Our primary inference is an interval estimate of the paired difference in squared error between CGAN-KSR and the KSR baseline across the eight folds. A percentile bootstrap with 20 000 resamples gives $\Delta\text{MSE} = 0.2208$ with a 95% confidence interval of $[0.023, 0.443]$,

Table 5. LOOCV results on experimental data ($n = 8$) under the leakage-free protocol (Algorithm 1). Every model is evaluated in the same eight folds. Models in the upper block are trained on the seven real samples of each fold; models in the middle block receive the identical augmented set of 7 real + 20 CGAN-generated samples ($z_{dim} = 4$, 3000 epochs). The lower block contains the two ablations. R_a observed range: 0.196–1.789 μm .

Model	n_{train}	MSE	RMSE (μm)	MAE
<i>Trained on real samples only</i>				
KSR baseline	7	0.3632	0.6026	0.5182
Linear regression [33]	7	0.3771	0.6141	0.5160
SVR (RBF) [34]	7	0.2658	0.5156	0.4175
Random Forest [35]	7	0.2745	0.5239	0.4384
Gaussian Process (Matérn 5/2) [11, 13]	7	0.3521	0.5934	0.4889
KSR + bootstrap [36]	7	0.3217	0.5672	0.4936
RSM, power law $Cs^{b_1}v^{b_2}t^{b_3}$	7	0.4483	0.6695	0.5056
RSM, first order + interactions [†]	7	0.1404	0.3747	0.3747
<i>Trained on 7 real + 20 CGAN-generated samples</i>				
KSR (proposed)	27	0.1424	0.3774	0.3180
SVR (RBF), augmented	27	0.1473	0.3838	0.3214
Gaussian Process, augmented	27	0.1531	0.3913	0.3435
Random Forest, augmented	27	0.1705	0.4129	0.3411
Linear regression, augmented	27	0.2350	0.4847	0.3854
<i>Ablations</i>				
KSR + interpolation augmentation	27	0.3562	0.5968	0.5123
KSR + unconditional noise augmentation	27	0.5816	0.7626	—

Bold row = proposed method. Noise ablation: mean over 20 independent draws (sd 0.199). Bootstrap baseline: $B = 50$ resamples drawn within each fold; no new conditions. [†]Saturated on this design (7 parameters, 7 training points); see the discussion below.

which excludes zero; the paired effect size is Cohen's $d_z = 0.684$. We designate this comparison as primary and treat all others as descriptive, in preference to applying a multiplicity correction to a set of non-independent tests at this sample size.

We report the secondary tests together with their limitations rather than selecting among them. The augmented model attains lower squared error in 6 of the 8 folds; the one-sided exact sign test gives $p = 0.145$, and the Wilcoxon signed-rank test gives $p = 0.074$. Neither reaches conventional significance, and we make no claim of statistical significance on the basis of the sign test. This should be read in light of a structural limitation rather than as evidence of no effect: at $n = 8$ the one-sided sign test cannot attain $p < 0.05$ unless at least 7 of 8 folds improve, its attainable p -values being 0.363, 0.145, 0.035 and 0.004 for 5, 6, 7 and 8 improved folds respectively. The test assumes only independence of fold outcomes; the Wilcoxon test additionally assumes symmetry of the paired differences, which cannot be verified at this sample size and is therefore flagged.

That independence assumption deserves scrutiny in its own right, and applies equally to the bootstrap interval above: the eight LOOCV folds are not eight independent replications of the experiment. Each fold's training set shares seven of the eight original samples with every other fold's training set, differing only in which single point is withheld, so the eight per-fold models—and the eight

paired differences on which every test in this subsection is based—are estimated from substantially overlapping data rather than independent draws. We continue to report LOOCV as the primary protocol because $n = 8$ admits no alternative that does not discard observations the design cannot spare, but on this basis we treat the interval, the effect size and both secondary tests as supportive, exploratory evidence of a consistent directional effect, not as confirmatory evidence with the sampling properties of eight independent trials.

Folds 6 and 8 are the two cases where the baseline outperforms the augmented model. Fold 7 merits comment in the opposite direction: its squared error is 10^{-4} , essentially exact. This is a consequence of the corrected protocol rather than exceptional accuracy. Because a synthetic sample now lies exactly at the held-out coordinate, it dominates the local kernel weighting, and the augmented prediction closely tracks the generator output there. Section 3.4 quantifies this effect.

The bootstrap baseline (0.3217) resamples the seven real observations without introducing any new cutting condition, and improves on the plain baseline only marginally. Together with the interpolation ablation this delimits the source of the gain: neither resampling existing conditions nor populating new ones with uninformative values is sufficient.

Effect of the grid-filtering protocol, and why correcting it improved accuracy. An earlier version of this work removed

all eight experimental coordinates from the candidate grid once, before the cross-validation loop, leaving 19 generation points per fold. That is defective: excluding the held-out coordinate uses knowledge of where the held-out sample lies, even though its measured R_a is never seen. The protocol used throughout this paper removes only the seven training coordinates within each fold, leaving 20 generation points, one of which coincides with the held-out coordinate. We report both, since the choice changes the conclusions in opposite directions:

Grid filtering	MSE	RMSE (μm)	Folds improved	Sign test p
Global, before loop (19 pts)	0.1912	0.4373	7 of 8	0.035
Per fold (20 pts), used here	0.1424	0.3774	6 of 8	0.145

Correcting the protocol lowered the error, which is the opposite of the usual direction for a leakage fix and therefore requires explanation. The reason is that the defective filter withheld the one generation point that matters most for the prediction being made: the held-out coordinate itself. Restoring it lets the generator place a synthetic sample where the estimate is required, which sharpens the point prediction, and this is legitimate because the generator is trained on the seven training responses alone. The accompanying cost is that the sign test now favours the augmented model in 6 folds rather than 7, so the improvement no longer reaches significance by that test. We regard the per-fold protocol as the correct one on grounds of principle rather than of outcome, and we present the comparison so that the effect of the choice is visible rather than implicit.

3.3. Response-Surface Stability and Decision Usefulness

Beyond scalar error metrics, response-surface stability is essential for cutting-parameter selection. Figs. 4 and 5 show that augmentation produces surfaces with reduced local variance (measured by the mean absolute surface gradient, MAG; see below), particularly in regions sparsely supported by the original $2 \times 2 \times 2$ experimental design.

Surface smoothness is quantified by the Mean Absolute Gradient (MAG), computed from the 40×40 prediction grids in Figs. 4 and 5: $\text{MAG}(\text{KSR augmented}) < \text{MAG}(\text{KSR baseline})$ across all three cross-sectional planes: plane $t = 1$ mm: 0.0066 vs. 0.0101 ($\downarrow 34.3\%$); plane $s = 60$ mm/min: 0.0122 vs. 0.0191 ($\downarrow 36.1\%$); plane $v = 1000$ rpm: 0.0057 vs. 0.0078 ($\downarrow 27.0\%$). The overall figure, 32.5%, is the unweighted mean of these three per-plane reductions computed from the full-precision (unrounded) MAG values; averaging the values as displayed to one decimal place gives the same 32.5% after rounding.

This is the specific role kernel smoothing plays in the

framework, and the basis on which we retain it. As Table 5 shows, an augmented SVR or Gaussian Process predicts about as accurately, so KSR is not adopted for predictive superiority. Its value lies in the deliverable: it converts 27 scattered points into a continuous surface with a single interpretable hyperparameter and without the axis-aligned discontinuities a tree ensemble produces, so that a process window can be read directly from Fig. 5.

Physical interpretation of the observed trends. The measured roughness cannot be explained by feed-mark kinematics. For a corner radius $r_e = 0.8$ mm the classical estimate $R_a \approx f_z^2 / (32r_e)$ predicts 0.007–0.098 μm , whereas the measurements span 0.196–1.789 μm —larger by factors of 2 to 258, and negatively correlated with the kinematic prediction ($r = -0.51$). Both counter-intuitive trends, roughness improving with feed and degrading with spindle speed, follow instead from the minimum chip thickness $h_{\min}$: since $f_z = s / (nz)$, raising the table feed increases f_z while raising the spindle speed decreases it. The two conditions with the lowest feed per tooth ($f_z = 13.3$ μm , runs 3 and 7) are exactly the two with the highest roughness, averaging 1.657 μm against 0.555 μm for the six conditions with $f_z \geq 20$ μm . Reported $h_{\min}$ for conventionally ground carbide edges is of order 20 μm [30], so these two conditions plausibly operate below the threshold for clean chip formation, with ploughing rather than shearing dominating.

This is a physically grounded hypothesis consistent with the data, not a demonstrated mechanism: at $n = 8$, f_z is collinear with the (low feed, high speed) combination, so an $h_{\min}$ effect cannot be separated statistically from a feed-speed interaction ($r = -0.587$, $p = 0.13$). Confirming it would require measurements at intermediate feeds and an edge-radius measurement on the inserts used. We note only that it is better supported than the chip-morphology conjecture offered previously, which we withdraw.

3.4. Additional Analyses and Sanity Checks

Training stability. Per-epoch loss curves for all eight folds, together with the complete 8×3000 loss history, are provided as supplementary material (Fig. S1 and Data S3). Averaged over the final 1000 epochs, the discriminator loss is 0.6933 ± 0.0084 and the generator loss 0.6943 ± 0.0126 —every fold lies within 0.0008 of the empirical convergence criterion defined in Section 2.5, $-\ln 0.5 = 0.6931$. A previous version of this work quoted $D_{\text{loss}} \approx 0.35$ –0.45 and described it as near equilibrium; that statement was erroneous and is withdrawn. We stress that equilibrium loss values are a necessary but not a sufficient condition for a faithful generator—an uninformative discriminator is equally consistent with a good generator and with two

weak networks—so we treat the curves as evidence against divergence and mode collapse only, and rely on the ablations of Section 3.1 for evidence that the samples carry conditional information.

Conditional dispersion. With $M = 10$ independent realizations per condition, the average conditional standard deviation is $\sigma_{\text{cgan}} = 0.030 \mu\text{m}$, about 1.9% of the experimental R_d range (Table 4). This is non-zero, so the generator has not collapsed to a deterministic map, but it is small enough that the generator behaves close to a conditional mean predictor—a point taken up below.

Distributional comparison. A two-sample Kolmogorov-Smirnov test between real and generated R_d gives $D = 0.4342$, $p = 0.178$. We do not interpret this as evidence that the distributions agree. The statistic is substantial, and at $n = 8$ the test has negligible power against any but the grossest distributional differences, so a non-significant result here is uninformative. The claim that the synthetic distribution is “not significantly different” from the experimental one, made in the previous version, is withdrawn.

What the KSR step contributes. Because the corrected protocol places a synthetic sample at the held-out coordinate, it is necessary to ask how much of the augmented prediction is the generator’s output and how much is the smoothing. Comparing the raw generated value at the held-out coordinate with the final KSR prediction across the eight folds gives a Pearson correlation of $r = 0.991$ and a mean absolute difference of $0.066 \mu\text{m}$. Using the raw generator output directly as the predictor yields $MSE = 0.1582$ against 0.1424 for CGAN-KSR; the paired difference is $\Delta MSE = 0.0158$ with 95% CI $[-0.007, 0.041]$, which contains zero (Wilcoxon $p = 0.23$, 4 of 8 folds favouring the smoother). We therefore do not claim that the KSR step improves predictive accuracy. The accuracy gain reported in this paper is attributable to the conditional generator; the contribution of kernel smoothing is the smooth, readable response surface quantified in Section 3.3.

Sensitivity to the augmentation ratio. Varying the number of synthetic samples per fold over $[5, 10, 15, 20]$, with 10 independent random subsets at each level, gives $MSE = 0.2483 \pm 0.0948$, 0.1753 ± 0.0298 , 0.1686 ± 0.0181 and 0.1424 , against 0.3632 without augmentation. Performance improves monotonically with neither saturation nor degradation at the highest ratio, so the results do not depend critically on the choice of 20.

Sensitivity to the input parameterisation. Replacing the machine parameters (s, ν, t) by the physically meaningful (f_z, V_c, t) is a bijection, so it adds no information, but it

changes the metric in which the kernel measures proximity. CGAN-KSR is unaffected (0.1447 versus 0.1424; ΔMSE 95% CI $[-0.092, 0.082]$) while the baseline degrades to 0.5585, because the 2^3 design is orthogonal in (s, ν) by construction and the nonlinear map collapses it toward a curve. This is a third demonstration—alongside the regressor and ratio analyses—that the benefit resides in the augmentation rather than in a particular modelling choice. Details are given in the supplementary material.

3.5. Uncertainty Calibration under Augmentation

Point estimates alone are inadequate at this sample size, so we examined the predictive distributions produced by Gaussian Process Regression, which supplies a principled posterior variance, in both the real-only and augmented regimes.

GP specification, for reproducibility. The kernel is fixed a priori, not tuned against the evaluation folds: a Matérn kernel ($\nu = 2.5$, initial length scale 0.5, bounds $[10^{-2}, 10^2]$) with an output-scale prefactor (bounds $[10^{-3}, 10^3]$), plus an additive white-noise term representing observation noise (initial value 10^{-2} , bounds $[10^{-6}, 10^1]$). Only these bounds, not the kernel family or the initial values, are the same for every fit. Hyperparameters are optimised by maximising the log marginal likelihood (L-BFGS, 10 random restarts) independently within each of the eight LOOCV folds and separately for the real-only and augmented regimes—16 independent fits in total, none sharing information across folds or between regimes. The target is internally standardised before fitting. The predictive distribution at the held-out condition is Gaussian with mean μ and standard deviation σ returned by the fitted GP; the reported 95% prediction interval is $\mu \pm 1.96\sigma$, and coverage is the fraction of the eight held-out measurements falling inside it.

The result is a caution rather than a reassurance. On the seven real samples the GP is close to nominally calibrated: mean predictive standard deviation $0.504 \mu\text{m}$, and 95% prediction intervals covering 7 of 8 held-out cases (88%). After augmentation the point accuracy improves markedly ($MSE 0.3521 \rightarrow 0.1531$) but the intervals narrow almost sevenfold, to $0.074 \mu\text{m}$, while their coverage collapses to 2 of 8 cases (25%) against a nominal 95%. In 6 of the 16 fits the optimizer drove the noise term to its lower bound, i.e. the GP interpolates a self-consistent synthetic surface.

The mechanism is straightforward: the 20 synthetic responses are treated as independent observations, so the posterior contracts as though 27 measurements existed, whereas they carry no independent information about the true response. Augmentation buys accuracy at the cost of calibration, and manufactures confidence that is not

warranted. This is the empirical form of the failure mode anticipated in Section 2.7: a biased conditional mean is inherited by the augmented estimator together with a false appearance of precision, and it is not detectable from the augmented data alone.

The practical consequence is explicit: predictive intervals obtained from an augmented model must not be used to set process tolerances or acceptance limits. Where uncertainty matters for a decision, it should be estimated from the real observations alone, accepting the wider and more honest intervals that result.

3.6. Boundary of Model Validity

We state the conditions under which the framework may be applied, and those under which it may not.

- **Material, tool and range.** The evidence covers Y8 carbon tool steel, face milling with a $\varnothing 25$ mm three-insert coated-carbide cutter, and $s \in [60, 150]$ mm/min, $n \in [1000, 1500]$ rpm, $t \in [0.2, 1.0]$ mm. Machining dynamics and surface-formation mechanisms differ substantially across material-tool pairings, and nothing in our evidence extends beyond this window.
- **Transferability.** The trained generator weights are specific to this pairing and are not transferable without retraining. Only the procedure transfers; the model does not.
- **Extrapolation.** Outside the sampled ranges the kernel estimator degenerates toward the local mean and the generator is queried away from its training support. Neither behaviour is validated here.
- **Absolute accuracy.** RMSE is $0.377 \mu\text{m}$ against a mean measured R_a of $0.830 \mu\text{m}$, i.e. approximately 45% of the mean response. Expressed against the roughness grade numbers still widely used on engineering drawings, the measured surfaces span roughly N4 (R_a $0.2 \mu\text{m}$) to N7 (R_a $1.6 \mu\text{m}$), and the model's RMSE is comparable to a full grade step (N5→N6, 0.4 to $0.8 \mu\text{m}$). The framework can therefore support coarse screening of a process window, but it cannot reliably resolve adjacent roughness grades and must not be relied on to certify a finish specification. We state this explicitly because a 60.8% relative improvement could otherwise be mistaken for absolute adequacy.
- **Physical fidelity versus predictive performance.** The LOOCV protocol validates predictive performance only at the eight experimentally observed conditions, each evaluated once as the held-out point. It says nothing about the physical accuracy of the generated R_a values

at the remaining interior grid conditions—the majority of the 27-point grid underlying the response surfaces of Fig. 5—since no measurement exists there against which to check them. Improved LOOCV accuracy at the observed conditions is evidence that the generator's outputs are informative where they can be verified; it is not evidence that the response surface built from all 27 points is physically accurate where they cannot be. Claims about predictive performance and claims about the physical validity of the generated response surface are kept separate throughout this paper, and the latter is not established by the present evidence.

- **Training stochasticity.** The seed policy in Section 2.5 ($42 + i$) gives each fold a distinct, reproducible seed but does not by itself quantify training-to-training variability. An independent-retraining check—ten CGAN trainings per fold with seeds unrelated to $42 + i$; full protocol and per-fold results in the supplementary material—gives a mean augmented MSE of 0.192 ± 0.036 , against the reported 0.1424 and the non-augmented baseline of 0.363: still an approximately 47% reduction, but the single-seed headline figure lies toward the favourable end of the retraining distribution rather than at its centre, and individual folds vary considerably in whether that particular seed was favourable or unfavourable. The reported 0.1424 should be read as the outcome of the pre-specified evaluated configuration, not as the expected performance of a freshly retrained model.
- **Uncertainty.** As shown in Section 3.5, predictive intervals from augmented models are not trustworthy.
- **Uncontrolled factors.** Temperature, machine vibration and coolant flow were not monitored, and no cutting condition was repeated, so run-to-run process variability is not separable from measurement variability in these data.

The study therefore demonstrates feasibility under data scarcity, not general applicability.

3.7. Computational Cost and Industrial Deployment

On an Intel Core i7-9850H (6 cores, 2.60 GHz), 32 GB RAM, CPU-only PyTorch, training the CGAN for 3000 epochs takes 9.5 ± 1.0 s per fold, generation of the 20 synthetic samples 5 ms, and KSR bandwidth selection with prediction 7 ms. Building one deployable model therefore costs about 9.5 s and querying it 7 ms; the full eight-fold evaluation takes 76 s. A Gaussian Process on the seven real samples fits in 0.26 s, so the augmented pipeline costs roughly 36 times more computation for a 60% reduction in MSE—and forfeits calibrated uncertainty.

These figures place the method as an offline process-planning aid, not an online monitor: model building is negligible against the machining time it informs, and inference is fast enough for interactive parameter selection. Retraining is required on any change of material, tool geometry, insert grade or machine. The formulation cannot adapt to tool wear, which is not among the conditioning variables and would require measurements across the tool life to incorporate.

For practitioners the workflow is: collect a $2 \times 2 \times 2$ factorial design; train the CGAN offline; generate synthetic samples at the intermediate grid conditions; fit KSR to the combined set; and query the surface within the experimental range, subject to the two limits stated above—coarse screening only, and no use of the model's own uncertainty estimates. We can delimit the useful regime only qualitatively, having studied a single value of n : augmentation should matter most where the design is too sparse for a local estimator to resolve interior structure and further experiments are cost-prohibitive, but we have not tested either boundary and assert no thresholds. Two further limitations compound those of Section 3.6: adversarial training on seven samples per fold remains intrinsically fragile, and errors in generated samples propagate into the fit with only partial damping by the kernel weighting.

4. Conclusions

As a worst-case feasibility study using a $2 \times 2 \times 2$ factorial design ($n = 8$), this paper proposes a data-efficient CGAN-KSR framework for surface roughness (R_a) prediction in CNC milling under ultra-small experimental datasets. CGAN is used to generate condition-consistent synthetic samples to densify sparsely observed regions of the cutting-parameter space, while KSR provides a lightweight local regressor to obtain smooth and stable response surfaces. In summarising what the study yields, we separate what the evidence supports from what it does not.

Supported by the present evidence. Under the leakage-free protocol ($n = 8$), augmentation reduces MSE from 0.3632 to 0.1424 and RMSE from 0.603 to $0.377 \mu\text{m}$; the bootstrap 95% interval for the paired MSE difference, $[0.023, 0.443]$, excludes zero ($d_z = 0.684$). This single-seed headline figure should be read alongside the independent-retraining estimate below (0.192 ± 0.036); both indicate an improvement over the non-augmented baseline (0.363), but the headline value lies toward the favourable end of the retraining distribution rather than at its centre.

The ablation design points to conditional generation as

the source of the gain rather than sample volume: augmenting the same conditions with interpolated values yields no detectable improvement, and unconditional noise is actively harmful. We treat this as evidence consistent with a causal role for conditioning under the ablations tested, not as a proof that excludes every alternative explanation.

It does not depend on the downstream regressor—augmented SVR (0.1473) and Gaussian Process (0.1531) match augmented KSR (0.1424), and all five regressors evaluated in both regimes improve under augmentation, by 38–61%—nor on the input parameterisation.

Kernel smoothing contributes the deliverable, not the accuracy: it reduces the mean absolute surface gradient by 32.5% but does not measurably improve predictive error over the raw generator output.

Adversarial training is stable at $n = 7$ here, all eight folds converging to within 0.0008 of the empirical convergence criterion, $D_{\text{loss}} \rightarrow -\ln 0.5$.

Not established. Statistical significance under the sign test ($p = 0.145$), where the interval estimate and the test disagree at this sample size; transfer to other materials, tools or machines, the generator weights being non-transferable; physical fidelity of the generated responses, unassessable without measurements at the intermediate conditions; adequacy for specification-grade prediction, the RMSE being comparable to a full roughness grade step; and trustworthy uncertainty, which augmentation degrades rather than improves.

The most informative next experiment is not a larger dataset but a small one placed deliberately: measuring R_a at the intermediate grid conditions would turn every generated value into a testable prediction and settle the fidelity question. Beyond that we intend to evaluate transfer across workpiece materials by a leave-one-material-out protocol, which tests generalisation on genuinely unseen conditions without new machining; incorporate physics-informed constraints such as the minimum-chip-thickness relation of Section 3.3; and develop augmentation schemes that preserve calibration, for instance by down-weighting synthetic samples in the likelihood rather than treating them as measurements.

Acknowledgments

The authors gratefully acknowledge the Advanced Machining and Mold Research Laboratory, 411M-C7, School of Mechanical Engineering, Hanoi University of Science and Technology, for providing research facilities, guidance, and technical support, and the industrial partner whose tool workshop hosted the machining trials.

Funding statement

This research was funded by Hanoi University of Science and Technology (HUST) under project number T2025-PC-033.

Author contributions

The authors confirm contribution to the paper as follows: Conceptualization, Anh-Dung Nguyen and Dinh-Khang Bach; Methodology, Anh-Dung Nguyen and Thai-Viet Dang; Software, Dinh-Khang Bach; Formal analysis, Thanh-Nhan Nguyen; Investigation and execution of the machining experiments and roughness measurements, Thanh-Nhan Nguyen; Resources, Anh-Dung Nguyen and Thai-Viet Dang; Data curation, Thanh-Nhan Nguyen; Writing—original draft preparation, Dinh-Khang Bach and Anh-Dung Nguyen; Writing—review & editing, Thanh-Nhan Nguyen and Thai-Viet Dang; Visualization, Anh-Dung Nguyen; Funding acquisition, Thanh-Nhan Nguyen; Supervision, Thai-Viet Dang; Project administration, Thanh-Nhan Nguyen. All authors have read and agreed to the published version of the manuscript.

The experimental campaign reported in Section 2.1 was conducted by Thanh-Nhan Nguyen and is documented in [30], of which he was the academic supervisor. The present paper contributes the modelling framework, the evaluation protocol and all analyses; it reports no new machining experiments.

Availability of data and materials

All data and code required to reproduce every number in this paper are provided.

Provenance. The experimental measurements were produced by the present authors and were first published in [30]; this paper reports no new machining trials. The thesis is cited wherever that record is drawn upon, so that the prior appearance of the data and photographs is on record. The complete raw record—three roughness traces for each of the eight cutting conditions, from which every reported R_a value is recomputed—is given in Table 3 and as supplementary Data S1, so that the averaging can be verified independently.

Code. The complete pipeline—CGAN training, the leakage-free LOOCV protocol, all baselines and ablations—together with two scripts that re-derive and cross-check every numerical value reported in this paper, is available from the corresponding author on request, and will be immediately available to the editor and reviewers should they wish to verify any result during assessment. It will be deposited in a public repository with a citable DOI upon acceptance.

The experimental data were generated at an industrial partner site, and open release of the dataset is subject to that partner's authorisation, which is being sought.

Supplementary material comprises: Data S1, the raw roughness measurements; Fig. S1 and Data S3, per-epoch generator and discriminator losses for all eight folds; Table S1, the KSR bandwidths selected in each fold for both the baseline and augmented models; and Table S2, the combined 27-sample dataset underlying the response surfaces.

Ethics approval

Not applicable. This study did not involve human participants or animal subjects.

Conflicts of interest

The authors declare no conflicts of interest to report regarding the present study. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- [1] J. H. Ko and C. Yin, (2026) "A review of artificial intelligence application for machining surface quality prediction: from key factors to model development" *Journal of Intelligent Manufacturing* 37: 775–798. DOI: [10.1007/s10845-025-02571-y](https://doi.org/10.1007/s10845-025-02571-y).
- [2] H. Yang, H. Zheng, and T. Zhang, (2024) "A review of artificial intelligent methods for machined surface roughness prediction" *Tribology International* 199: 109935. DOI: [10.1016/j.triboint.2024.109935](https://doi.org/10.1016/j.triboint.2024.109935).
- [3] J. V. Abellán-Nebot, C. Vila Pastor, and H. R. Siller, (2024) "A review of the factors influencing surface roughness in machining and their impact on sustainability" *Sustainability* 16(5): 1917. DOI: [10.3390/su16051917](https://doi.org/10.3390/su16051917).
- [4] L. Liu, X. Zhang, X. Wan, S. Zhou, and Z. Gao, (2022) "Digital twin-driven surface roughness prediction and process parameter adaptive optimization" *Advanced Engineering Informatics* 51: 101470. DOI: [10.1016/j.aei.2021.101470](https://doi.org/10.1016/j.aei.2021.101470).
- [5] L. Hu, R. Gao, and Y. B. Guo, (2023) "Transfer learning enhanced Gaussian process model for surface roughness prediction with small data" *Manufacturing Letters* 35: 1103–1108. DOI: [10.1016/j.mfglet.2023.08.094](https://doi.org/10.1016/j.mfglet.2023.08.094).

- [6] C. Cooper, J. Zhang, Y. B. Guo, and R. X. Gao, (2023) "Surface roughness prediction through GAN-synthesized power signal as a process signature" **Journal of Manufacturing Systems** 68: 660–669. DOI: [10.1016/j.jmsy.2023.05.016](https://doi.org/10.1016/j.jmsy.2023.05.016).
- [7] J. Wang, T. Chen, and D. Kong, (2023) "Knowledge-based neural network for surface roughness prediction of ball-end milling" **Mechanical Systems and Signal Processing** 194: 110282. DOI: [10.1016/j.ymssp.2023.110282](https://doi.org/10.1016/j.ymssp.2023.110282).
- [8] L. Sun, X. Huang, X. Wang, Y. Zhang, M. Li, and Z. Liu, (2026) "Physics-informed deep learning method for surface roughness prediction in milling processes" **Advanced Engineering Informatics** 69: 103803. DOI: [10.1016/j.aei.2025.103803](https://doi.org/10.1016/j.aei.2025.103803).
- [9] R. Wang, M. N. Cheng, Y. M. K. Loh, C. Wang, and C. F. Cheung, (2022) "Ensemble learning with a genetic algorithm for surface roughness prediction in multi-jet polishing" **Expert Systems with Applications** 207: 118024. DOI: [10.1016/j.eswa.2022.118024](https://doi.org/10.1016/j.eswa.2022.118024).
- [10] S. Pedrammehr, M. Hejazian, M. R. Chalak Qazani, H. Parvaz, S. Pakzad, M. Ettefagh, et al., (2022) "Machine learning-based modelling and meta-heuristic-based optimization of specific tool wear and surface roughness in the milling process" **Axioms** 11(9): 430. DOI: [10.3390/axioms11090430](https://doi.org/10.3390/axioms11090430).
- [11] J. Chen, J. Lin, M. Zhang, and Q. Lin, (2024) "Predicting surface roughness in turning complex-structured workpieces using vibration-signal-based Gaussian process regression" **Sensors** 24(7): 2117. DOI: [10.3390/s24072117](https://doi.org/10.3390/s24072117).
- [12] C. Feng, M. Li, H. Guo, B. Qiu, and J. Zhang, (2024) "Energy efficiency identification and surface roughness prediction using cutting force signal for computer numerical controlled machine systems" **Scientific Reports** 14(1): 19004. DOI: [10.1038/s41598-024-69979-z](https://doi.org/10.1038/s41598-024-69979-z).
- [13] L. Song, C. Yan, G. Tu, M. Xiang, and Y. Liu, (2023) "Prediction and optimization of surface roughness in high-speed dry milling of 30CrMnSiNiA using GPR and MO-HHO algorithm" **International Journal of Advanced Manufacturing Technology** 128: 4357–4377. DOI: [10.1007/s00170-023-12167-6](https://doi.org/10.1007/s00170-023-12167-6).
- [14] C. Cooper, J. Zhang, Y. B. Guo, and R. X. Gao, (2025) "Surface roughness prediction in machining using two-stage domain-incremental learning with input dimensionality expansion" **Journal of Manufacturing Systems** 80: 503–510. DOI: [10.1016/j.jmsy.2025.03.014](https://doi.org/10.1016/j.jmsy.2025.03.014).
- [15] A. Balasuadhakar, S. T. Kumaran, and M. Uthayakumar, (2025) "Machine learning prediction of surface roughness in sustainable machining of AISI H11 tool steel" **Smart Materials in Manufacturing** 3: 100075. DOI: [10.1016/j.smmf.2025.100075](https://doi.org/10.1016/j.smmf.2025.100075).
- [16] J. K. Abbas, S. H. Aghdeab, and A. Al-Habaibeh, (2025) "Investigating the effect of process parameters on surface roughness of AISI M2 steel in EDM using deep learning neural networks" **International Journal of Advanced Manufacturing Technology** 137: 251–262. DOI: [10.1007/s00170-025-15184-9](https://doi.org/10.1007/s00170-025-15184-9).
- [17] M. Guo, S. Wei, C. Han, W. Xia, C. Luo, and Z. Lin, (2024) "Prediction of surface roughness using deep learning and data augmentation" **Journal of Intelligent Manufacturing and Special Equipment** 5(1): 221–241. DOI: [10.1108/JIMSE-10-2023-0010](https://doi.org/10.1108/JIMSE-10-2023-0010).
- [18] B. Zhang, X. Liu, C. Yue, S. Liu, X. Li, S. Y. Liang, et al., (2025) "An imbalanced data learning approach for tool wear monitoring based on data augmentation" **Journal of Intelligent Manufacturing** 36: 399–420. DOI: [10.1007/s10845-023-02235-9](https://doi.org/10.1007/s10845-023-02235-9).
- [19] R. Liu and W. Tian, (2024) "A novel approach to surface roughness virtual sample generation to address the small sample size problem in ultra-precision machining" **Sensors** 24(11): 3621. DOI: [10.3390/s24113621](https://doi.org/10.3390/s24113621).
- [20] W. Tian, J. Zhang, F. Zhao, X. Feng, X. Mei, G. Chen, and H. Wang, (2024) "Interpolation-based virtual sample generation for surface roughness prediction" **Journal of Intelligent Manufacturing** 35: 343–353. DOI: [10.1007/s10845-022-02054-4](https://doi.org/10.1007/s10845-022-02054-4).
- [21] Y. Chen, Z. Yan, and Y. Zhu, (2024) "A comprehensive survey for generative data augmentation" **Neurocomputing** 600: 128167. DOI: [10.1016/j.neucom.2024.128167](https://doi.org/10.1016/j.neucom.2024.128167).
- [22] K. Sohn, H. Lee, and X. Yan. "Learning structured output representation using deep conditional generative models". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 28. Montreal, QC, Canada, 2015, 3483–3491. URL: https://papers.nips.cc/paper_files/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html.
- [23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. "Improved training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Long Beach, CA, USA, 2017, 5767–5777. DOI: [10.48550/arXiv.1704.00028](https://doi.org/10.48550/arXiv.1704.00028).

- [24] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Virtual, 2020, 6840–6851. DOI: [10.48550/arXiv.2006.11239](https://doi.org/10.48550/arXiv.2006.11239).
- [25] E. A. Nadaraya, (1964) "On estimating regression" **Theory of Probability and Its Applications** 9(1): 141–142. DOI: [10.1137/1109020](https://doi.org/10.1137/1109020).
- [26] G. S. Watson, (1964) "Smooth regression analysis" **Sankhya: The Indian Journal of Statistics, Series A** 26(4): 359–372. URL: <https://www.jstor.org/stable/25049340>.
- [27] L. Wasserman. *All of Nonparametric Statistics*. New York, NY, USA: Springer, 2006. DOI: [10.1007/0-387-30623-4](https://doi.org/10.1007/0-387-30623-4).
- [28] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2018. DOI: [10.1201/9781315140919](https://doi.org/10.1201/9781315140919).
- [29] A. Wibowo and M. I. Desa, (2012) "Kernel based regression and genetic algorithms for estimating cutting conditions of surface roughness in end milling machining process" **Expert Systems with Applications** 39(14): 11634–11641. DOI: [10.1016/j.eswa.2012.04.004](https://doi.org/10.1016/j.eswa.2012.04.004).
- [30] H. P. Tran. "Study on the influence of technological parameters on the machined surface quality of several steels used to manufacture tooling components in the metal-forming industry, by CNC milling". In Vietnamese; original title begins "Nghien cuu anh huong cua cac thong so cong nghe toi chat luong be mat gia cong" (diacritics omitted). Source of the experimental measurements reused in the present study. (Master's thesis). Hanoi, Vietnam: Hanoi University of Science and Technology, 2023. URL: <http://dlib.hust.edu.vn/handle/HUST/22858>.
- [31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al. "Generative adversarial nets". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Montreal, QC, Canada, 2014, 2672–2680. DOI: [10.48550/arXiv.1406.2661](https://doi.org/10.48550/arXiv.1406.2661).
- [32] M. Mirza and S. Osindero. *Conditional generative adversarial nets*. arXiv:1411.1784. 2014. DOI: [10.48550/arXiv.1411.1784](https://doi.org/10.48550/arXiv.1411.1784).
- [33] H. Öktem, (2009) "An integrated study of surface roughness for modelling and optimization of cutting parameters during end milling operation" **International Journal of Advanced Manufacturing Technology** 43: 852–861. DOI: [10.1007/s00170-008-1763-3](https://doi.org/10.1007/s00170-008-1763-3).
- [34] C. Duan and Q. Hao, (2014) "Surface roughness prediction of end milling process based on IPSO-LSSVM" **Journal of Advanced Mechanical Design, Systems, and Manufacturing** 8(3): JAMDSM0024. DOI: [10.1299/jamdsm.2014jamdsm0024](https://doi.org/10.1299/jamdsm.2014jamdsm0024).
- [35] G. A. V. B. Vasconcelos, M. B. Francisco, C. H. de Oliveira, E. L. Barbedo, L. G. Paes de Souza, and M. L. Noronha Motta Melo, (2024) "Prediction of surface roughness in duplex stainless steel top milling using machine learning techniques" **International Journal of Advanced Manufacturing Technology** 134: 2939–2953. DOI: [10.1007/s00170-024-14290-4](https://doi.org/10.1007/s00170-024-14290-4).
- [36] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Boca Raton, FL, USA: Chapman & Hall/CRC, 1994. DOI: [10.1201/9780429246593](https://doi.org/10.1201/9780429246593).