Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fast Prediction of CT Spatial Resolution from Acquisition Parameters and Image Patches Using Physics-Simulated Data and Machine Learning

Manya Gupta
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee https://orcid.org/0000-0001-8829-9980

Research Article

Keywords: computed tomography, modulation transfer function, slanted-edge method, surrogate model, Gaussian process, convolutional neural network

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11710412/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Fast Prediction of CT Spatial Resolution from Acquisition Parameters and Image Patches Using Physics-Simulated Data and Machine Learning

Manya Gupta[1], and Mayank Goswami [1, 2, #]
[1] Department of Physics, IIT Roorkee
[2]*Mehta Family School of Data Science and Artificial Intelligence*, IIT Roorkee,
Roorkee, Uttarakhand-247667, India.
[#]mayank.goswami@ph.iitr.ac.in

***Abstract*—The spatial resolution of a computed tomography (CT) scan, summarised by the frequency at which the modulation transfer function falls to one half (MTF50), depends jointly on focal-spot size, geometric magnification, detector blur and reconstruction voxel size. We built a simulation pipeline that scans an analytic slanted-edge and bar-pattern phantom at 400 acquisition settings and measures the ground-truth MTF with the edge-spread-function method and used it to train two kinds of fast estimators. A Gaussian-process surrogate predicts MTF50 from the four settings with a mean absolute error (MAE) of 0.106 lp/mm and $R^2 = 0.983$ on held-out settings, compared with 1.285 lp/mm for the classical quadrature blur formula and 0.293 lp/mm for a calibrated version of it, about 2,000 times faster than a single simulation. Adding physics-derived blur features improved every model and kept errors low in several extrapolation scenarios in which raw-feature models failed. A small residual convolutional network that sees only a 64 × 64 image patch reached 0.121 lp/mm per setting, close to the surrogate, but its error rose to 0.60 lp/mm on unseen generic phantoms, exposing a clear out-of-distribution limit. Within the studied ranges, magnification and detector blur were the most influential settings. Code, data and trained models are released.**

***Index Terms*—computed tomography, modulation transfer function, slanted-edge method, surrogate model, Gaussian process, convolutional neural network.**

I. Introduction

X-ray computed tomography (CT) rebuilds a cross-section of an object from hundreds of projections. How small a feature the final image can show — its spatial resolution — is set by the acquisition geometry and hardware: the size of the X-ray focal spot, the geometric magnification given by the source-to-object and source-to-detector distances, blur inside the detector, and the voxel size chosen for reconstruction. Resolution is usually reported through the modulation transfer function (MTF) and its 50 % point, MTF50, in line pairs per millimetre (lp/mm).

Choosing settings that resolve a required feature size — a crack in a battery electrode, a pore in bone — is a routine planning problem. Two tools exist. A quadrature formula that adds all blur widths is instantaneous but approximate, because it ignores sampling and reconstruction effects. A full simulation or a test scan of a resolution phantom is accurate but must be repeated for every candidate setting. When only an image is available and the settings are unknown, neither tool applies.

This work asks whether machine learning trained on physics-simulated data can be both fast and accurate. Our contributions are: (i) an open simulation pipeline and dataset of 400 acquisition settings with exact slanted-edge ground truth and 6,000 labelled image patches; (ii) a comparison of five tabular regressors and two physics baselines, including per-regime errors, bootstrap confidence intervals and extrapolation tests; (iii) an image-only convolutional estimator and an analysis of where it fails; and (iv) a sensitivity analysis of which setting limits resolution.

II. Theory

A. MTF and MTF50

A linear, shift-invariant imaging system blurs a point into the point spread function. Integrating it along one direction gives the line spread function (LSF); the edge spread function (ESF) is the response to a sharp step, and LSF = dESF/dx. The MTF is the normalised magnitude of the Fourier transform of the LSF:

$$\mathrm{MTF}(f) = |\mathscr{F}\{\mathrm{LSF}\}(f)| \,/\, |\mathscr{F}\{\mathrm{LSF}\}(0)| \tag{1}$$

MTF50 is the frequency at which $\mathrm{MTF}(f) = 0.5$; a higher value means a sharper system.

B. Slanted-edge method

If a straight edge is tilted by a few degrees relative to the pixel grid, the distances of different pixels to the edge fall at many sub-pixel offsets. Binning pixel values by distance at a quarter of the voxel size gives a four-times oversampled ESF, from which the LSF and MTF follow [1]. The finite-difference derivative and the binning each low-pass filter the result, with transfer functions $\mathrm{sinc}(2fb)$ and $\mathrm{sinc}(fb)$ for bin width b; we divide them out.

C. Blur sources referred to the object

For a source-to-detector distance SDD and source-to-object distance SOD, the magnification is $M = \mathrm{SDD}/\mathrm{SOD}$. A focal spot of standard deviation σ_f (FWHM/2.355) projects a penumbra which, scaled back to the object, has width

$$\sigma_{f,\mathrm{obj}} = \sigma_f\,(M-1)/M, \qquad \sigma_{d,\mathrm{obj}} = \sigma_d/M, \tag{2}$$

where σ_d is the detector blur; the detector pixel pitch p appears in the object as an aperture of width p/M. Raising M shrinks detector blur but enlarges focal-spot blur, so an optimum exists. The classical approximation treats every term as a Gaussian and adds variances, a box of width w contributing $w^2/12$, with voxel size v:

$$\sigma^2 = \sigma_{f,\mathrm{obj}}^2 + \sigma_{d,\mathrm{obj}}^2 + (p/M)^2/12 + v^2/12 \tag{3}$$

$$\mathrm{MTF50} \approx \sqrt{(\ln 2 \,/\, 2\pi^2)} \,/\, \sigma \approx 0.1874 \,/\, \sigma \tag{4}$$

Equation (4) is our physics-only baseline.

III. Related Work

The edge method for presampled MTF measurement was established for digital radiography by Samei *et al.* [1] and is the basis of the slanted-edge procedure used here. Friedman *et al.* [2] showed how CT MTF and noise power can be measured with the ACR accreditation phantom, and Tominaga *et al.* [3] proposed a tilted-wire method that remains reliable at very low contrast and high noise. Tilley *et al.* [4] modelled focal-spot blur, detector blur and correlated noise explicitly inside model-based reconstruction for flat-panel cone-beam CT, which motivates the separate blur terms in our simulator. Closest in spirit to our image estimator, Bauer *et al.* [5] trained a convolutional network to estimate the MTF of camera lenses from image content. Our reconstruction follows filtered back-projection [6], which is the central-slice form of the Feldkamp–Davis–Kress (FDK) algorithm [7]. We combine these ingredients into a CT-specific surrogate trained on exact simulated ground truth and evaluate it, together with an image-only estimator, against physics baselines.

IV. Method

A. Simulated acquisition

The phantom is analytic: a 5.5 mm-radius cylinder ($\mu = 0.10$ mm^{-1}) containing a 3 mm square insert tilted by 5° ($\Delta\mu = 0.10$ mm^{-1}) for the slanted edge, a 2.4 mm round insert ($\Delta\mu = 0.06$ mm^{-1}) and two rows of three-bar groups at 1–10 lp/mm (Fig. 1). For each setting we compute exact line integrals for 900 views over 180°, convert them to intensities, blur the intensities with the combined focal-spot and detector Gaussian of (2), integrate over the pixel aperture p/M ($p = 0.1$ mm), add Poisson noise for $I_0 = 10^5$ photons per pixel, take the logarithm, rebin to the voxel pitch and reconstruct a 12.8 mm field of view by Ram-Lak filtered back-projection. Blur is applied to intensities, not line integrals, because it physically acts on photons. The simulation is a 2-D central slice of a cone-beam geometry; the whole pipeline uses only NumPy and SciPy.

TABLE I
ACQUISITION PARAMETERS

Parameter	Range	Sampling
Focal-spot FWHM	0.005–0.20 mm	log
Magnification M	1.2–5.0	linear
Detector blur σ_d	0.02–0.15 mm	linear
Voxel size v	0.025–0.10 mm	linear
Fixed: SDD 600 mm, pitch 0.1 mm, 900 views, $I_0 = 10^5$		

SOD = SDD/M. 81 settings on a 3-level grid plus 319 Latin-hypercube samples.

B. Settings and data split

Table I lists the four varied settings. We combined a $3^4 = 81$-point grid with 319 Latin-hypercube samples (400 settings) and split them 70/15/15 by setting into training (280), validation (60) and testing (60), so no setting appears in more than one split. For stratified reporting, magnification and focal spot were each divided into three equal bins of their sampling range.

C. Ground-truth MTF

Both near-vertical edges of the square insert are used. Pixels within ±1.0 mm across and ±0.9 mm along an edge are binned by their exact analytic distance to the edge at $v/4$, the LSF is windowed (Tukey, $\alpha = 0.4$), and MTF50 and MTF10 are read by linear interpolation. Because the edge geometry is known exactly, no edge fitting is needed — a common source of error in the method. Before generating data, automated checks confirmed that the projector reproduces exact chord lengths, that synthetic Gaussian edges give MTF50 within 3 % of the analytic value $0.1874/\sigma$, and that reconstructed attenuation values are correct in value and position. All 400 settings passed quality control (edge contrast within ±20 % of the design value), and MTF50 spans 1.6–8.3 lp/mm (Fig. 1).

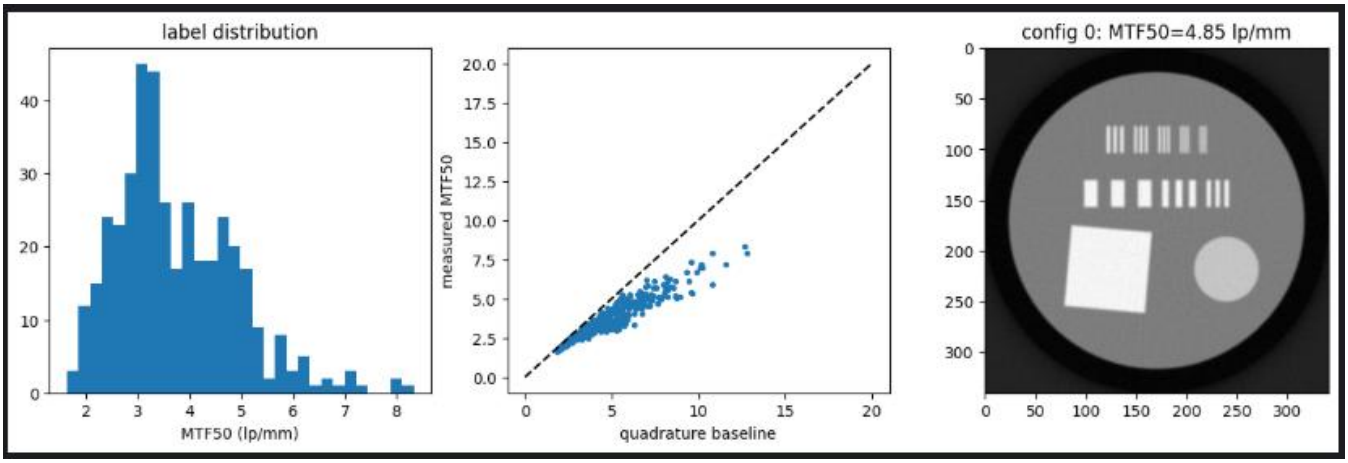


Fig. 1. Dataset overview. Left: distribution of ground-truth MTF50 over the 400 settings. Centre: measured MTF50 against the quadrature formula (4); the formula is systematically optimistic. Right: reconstructed phantom for one setting, showing the bar groups, the 5°-tilted square used for the slanted edge, and the round insert.

D. Tabular surrogate

Two feature sets were used: *raw* (the four settings) and *physics-informed* (raw plus $\sigma_{f,obj}$, $\sigma_{d,obj}$, p/M and the formula value (4)). Inputs were z-scored. The models, implemented with scikit-learn [8] and XGBoost [9], were linear regression; a Random Forest; gradient-boosted trees (XGBoost); a multilayer perceptron with two hidden layers of 64 ReLU units trained with Adam (learning rate 10^{-3}, batch 32, 200 epochs, MSE loss); and a Gaussian process (GP) [10] with an automatic-relevance RBF or Matérn-5/2 kernel plus white noise. Hyper-parameters were chosen by grid search on the validation split only; each chosen model was then refitted with five seeds. Two physics baselines need no training: the formula (4), and a calibrated formula with a straight-line correction

fitted on the training split. Test uncertainty is given as 95 % bootstrap confidence intervals over the 60 test settings.

E. Image-based estimator

Every reconstruction was resampled to a common 0.025 mm grid so that a 64 × 64 patch always covers 1.6 × 1.6 mm, and normalised so that background is 0 and the edge insert is 1. Fifteen patches per setting (square edges, round edge and bars, jittered by ±0.3 mm) inherit the setting's MTF50, giving 4,200/900/900 patches. The network is a ResNet-lite [11]: three residual blocks (16, 32, 64 channels, each two 3 × 3 convolution–batch-norm–ReLU layers with a 1 × 1 shortcut and 2 × 2 max-pooling), global average pooling and a 64–32–1 head (76,929 parameters). It was trained with Adam (10^{-3}), batch 32, MSE on standardised targets for 60 epochs, keeping the epoch with lowest validation MAE; three seeds were trained, taking about 1.5 min each on a T4 GPU. Augmentation was restricted to flips and 90° rotations, because arbitrary rotations require interpolation, which itself blurs the patch and would corrupt the label. A setting's prediction is the mean over its 15 patches.

F. Further experiments

Sensitivity was measured by permutation importance (30 repeats on the test split), one- and two-dimensional partial dependence, and GP length scales. Speed was measured on the same Colab CPU for the full simulation plus MTF measurement (five test settings) and for model inference on one setting and on a batch of 10,000. For extrapolation, models were trained on settings whose parameter lay in the lower two-thirds of its range and tested on the upper third, for each parameter and for magnification and focal spot jointly. Finally, the CNN was slid over 20 random-ellipse phantoms (step 0.4 mm), keeping windows inside the object that contain structure, to produce local-resolution maps.

V. Results

A. Tabular surrogate

Table II and Fig. 2 summarise the test results. The GP was selected on validation (raw features, validation MAE 0.088 lp/mm, versus 0.089 with physics features) and reached a test MAE of 0.106 lp/mm (95 % CI 0.078–0.140) with $R^2 = 0.983$, meeting the target of $R^2 \geq 0.95$ and MAE ≤ 0.3 lp/mm. The physics formula ranks settings well (Pearson 0.952) but overestimates MTF50 so strongly that R^2 is negative (−0.60); a single linear correction reduces its error to 0.293 lp/mm, and learning reduces it by a further factor of about three. Linear regression on raw settings (MAE 0.374, $R^2 = 0.81$) confirms that the relationship is non-linear. Physics-informed features improved every model, most strongly the linear model (0.374 → 0.179) and the Random Forest (0.225 → 0.142). Because the GP and the linear model are deterministic, their five seeds are identical; their uncertainty is captured by the bootstrap intervals.

TABLE II
TABULAR MODELS ON THE TEST SPLIT

Model (features)	MAE [95 % CI]	RMSE	R^2
GP (physics)	0.087 [0.056–0.127]	0.166	0.983
GP (raw) ‡	0.106 [0.078–0.140]	0.164	0.983
MLP (physics)	0.113 [0.073–0.148]	0.187	0.978
MLP (raw)	0.125 [0.080–0.158]	0.200	0.975
XGBoost (physics)	0.132 [0.093–0.179]	0.214	0.972
Random Forest (physics)	0.142 [0.102–0.189]	0.227	0.968
XGBoost (raw)	0.177 [0.118–0.240]	0.288	0.949
Linear (physics)	0.179 [0.129–0.244]	0.294	0.947
Random Forest (raw)	0.225 [0.163–0.299]	0.350	0.924
Calibrated formula	0.293 [0.229–0.363]	0.392	0.905
Linear (raw)	0.374 [0.280–0.487]	0.554	0.810
Quadrature formula	1.285 [1.063–1.533]	1.606	−0.596

MAE and RMSE in lp/mm, mean over five seeds; CI from 2,000 bootstrap resamples of the 60 test settings. ‡ selected on the validation split.

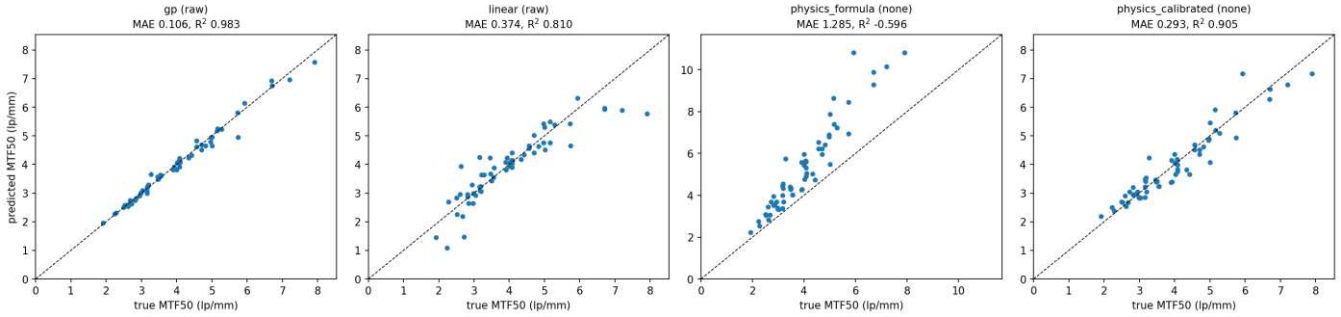


Fig. 2. Predicted against true MTF50 on the 60 test settings for the selected GP, linear regression, the quadrature formula and the calibrated formula. Note the different vertical scale for the formula, which overestimates almost every setting.

B. Error across operating regimes

The GP error was uniform across the operating range: 0.066, 0.134 and 0.108 lp/mm for low, mid and high magnification, and 0.140, 0.101 and 0.071 lp/mm for small, medium and large focal spots (15–26 test settings per bin). The linear model ranged from 0.30 to 0.48 lp/mm and the calibrated formula from 0.18 to 0.41 lp/mm. The largest GP error occurs for small focal spots, where several other blur terms compete.

C. Example MTF curves

Fig. 3 compares measured MTF curves for four test settings with Gaussian-shaped curves, $\mathrm{MTF}(f) = 0.5^{(f/\mathrm{MTF50})^2}$, through the predicted MTF50. Both estimators place the 50 % point correctly from 1.9 to 7.9 lp/mm; the measured curves show additional ripple at high frequency from noise and sampling that a single number cannot capture.

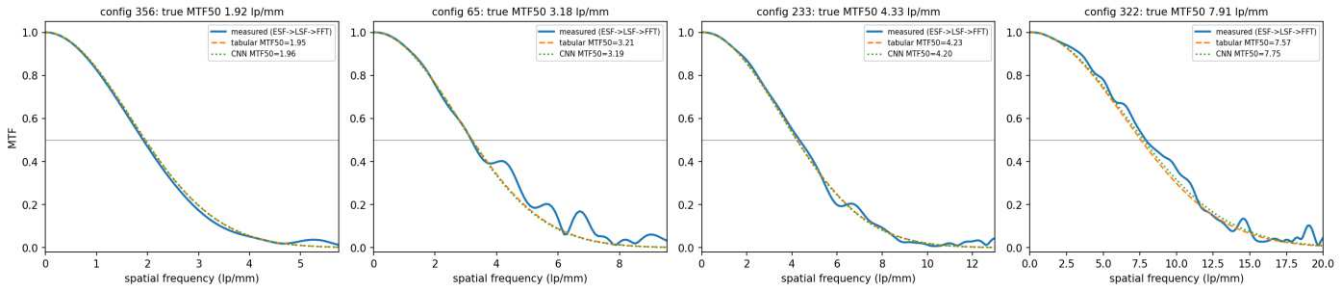


Fig. 3. Measured MTF (ESF → LSF → FFT) of four test settings spanning the MTF50 range, with Gaussian-shaped curves through the MTF50 predicted by the GP surrogate (dashed) and by the CNN (dotted).

D. Sensitivity

Permutation importance (Fig. 4) gives the same ranking for all four models: shuffling magnification raises the GP test MAE by 0.63 lp/mm, detector blur by 0.60, voxel size by 0.51 and focal spot by 0.36. The partial-dependence surfaces (Fig. 5) show the expected trade-off: increasing magnification raises MTF50 strongly when the focal spot is small, but for focal spots above about 0.12 mm the gain disappears and the contours become flat. The GP length scales rank voxel size as the fastest-changing input (2.6, against 4.0, 6.0 and 6.3 for magnification, focal spot and detector blur); length scales describe local rate of change on standardised inputs rather than total effect, so the two measures answer different questions.

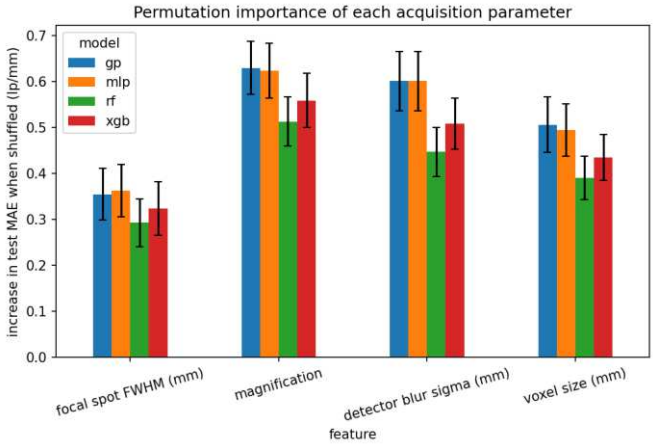


Fig. 4. Permutation importance: increase in test MAE when one acquisition parameter is randomly shuffled, for four models (mean ± s.d. over 30 repeats).

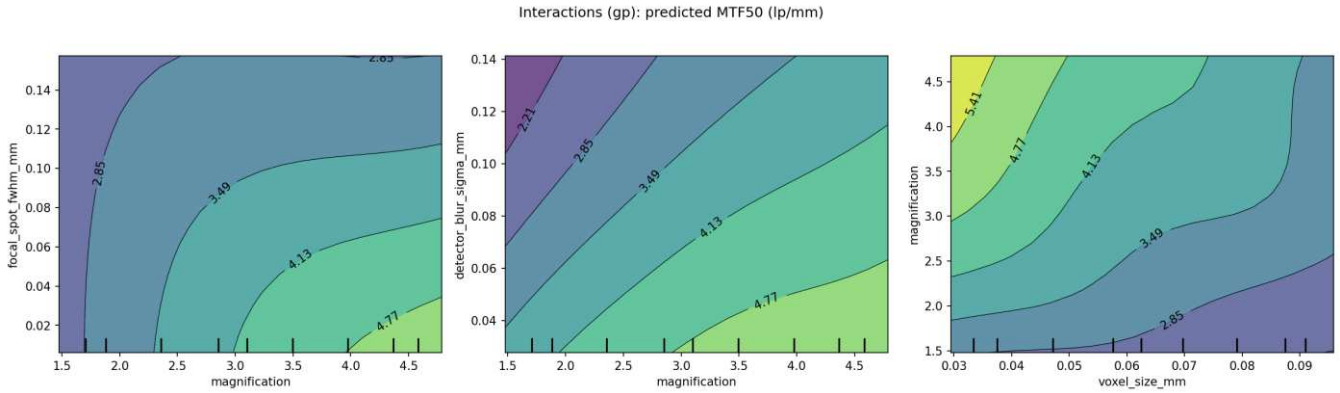


Fig. 5. Two-dimensional partial dependence of the GP prediction. Left: magnification × focal spot — higher magnification only helps when the focal spot is small. Centre: magnification × detector blur. Right: voxel size × magnification.

E. Speed

The full simulation with MTF measurement took 3.64 ± 1.14 s per setting on the Colab CPU (7.0 s on average on Kaggle during data generation). The GP needed 1.65 ms for one setting (about 2,200 × faster) and 22 µs per setting in a batch of 10,000 (about 1.7×10^5 ×); XGBoost and the MLP were similar. The Random Forest needed 0.24 s for a single query because of parallel-thread start-up, but only 56 µs per setting in a batch. Re-simulating the five timed settings reproduced the stored MTF50 values exactly, confirming that the pipeline is deterministic.

F. Extrapolation

Inside the training range all learned models reached R^2 between 0.78 and 0.99. Outside it (Table III, Fig. 6), raw-feature models degraded as expected: the GP fell to $R^2 = 0.64$ when magnification exceeded the training range and to −0.20 when magnification and focal spot did so together, and the tree models fell below zero in the focal-spot scenario (Random Forest −0.43, XGBoost −0.24), since trees cannot predict beyond the targets they have seen. With physics-informed features most models stayed accurate — the GP kept $R^2 = 0.97$ in the focal-spot scenario and 0.99 in the detector-blur scenario, and the Random Forest 0.96 in the joint scenario. Two exceptions matter: all models degraded when the voxel size was extrapolated (best $R^2 = 0.86$), and the physics-feature MLP failed in the focal-spot scenario ($R^2 = -0.48$). Ratios of outside to inside MAE below one (Fig. 6) do not indicate improvement; they occur where the outside settings are blurrier, so absolute errors are smaller, and should be read together with R^2.

TABLE III
EXTRAPOLATION: R^2 OUTSIDE THE TRAINING RANGE

Model (features)	Focal	Mag.	Det.	Voxel	M + F
GP (raw)	0.78	0.64	0.95	0.69	−0.20
GP (physics)	0.97	0.82	0.99	0.76	0.87
MLP (raw)	0.41	0.94	0.93	0.83	0.34
MLP (physics)	−0.48	0.95	0.95	0.86	0.74
XGBoost (raw)	−0.24	0.67	0.66	0.42	−0.43
XGBoost (physics)	0.98	0.90	0.97	0.76	0.95
Random Forest (raw)	−0.43	0.61	0.58	0.31	−0.86
Random Forest (physics)	0.97	0.90	0.96	0.75	0.96
Linear (raw)	0.32	0.55	0.68	0.49	−3.06
Linear (physics)	0.85	0.93	0.94	0.80	0.65
Calibrated formula	0.85	0.85	0.85	0.61	0.78

Training on the lower two-thirds of each parameter's range, testing on the upper third. M + F: magnification and focal spot both in their upper third.

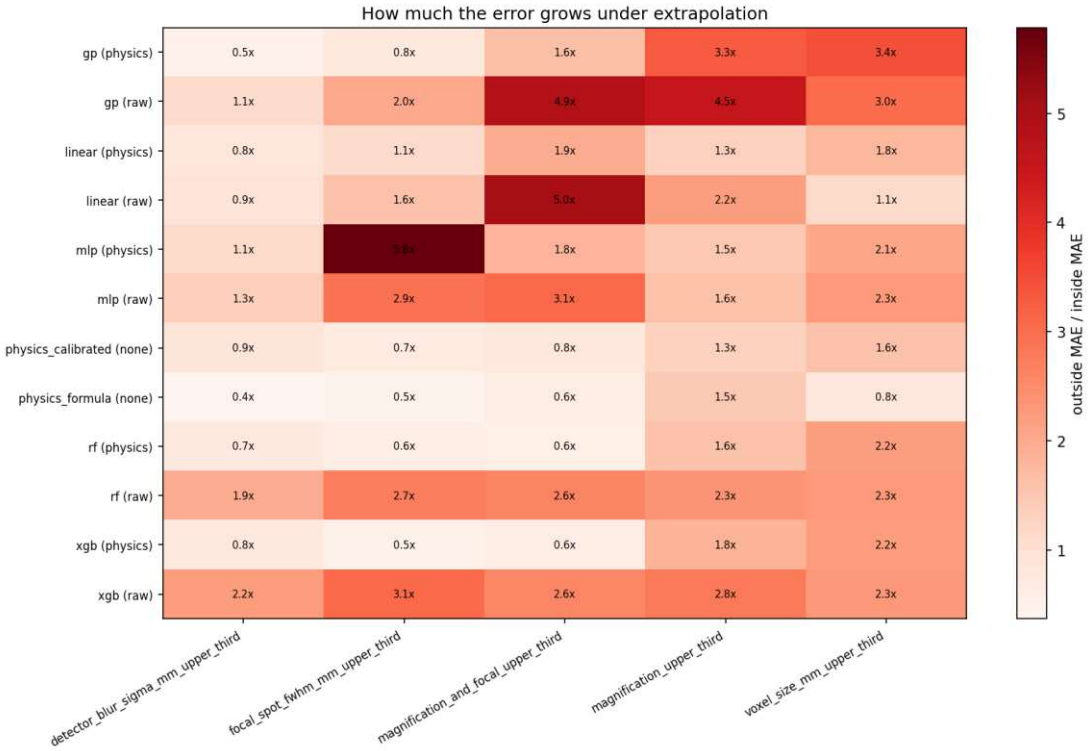


Fig. 6. Ratio of extrapolation MAE to interpolation MAE for every model and scenario. Physics-informed features (rows labelled "physics") keep the ratio near or below one in most scenarios; voxel-size extrapolation is hard for every model.

G. Image-based CNN estimator

The CNN reached a per-patch test MAE of 0.162 ± 0.010 lp/mm (R^2 = 0.955, three seeds) and a per-setting MAE of 0.121 ± 0.009 lp/mm (R^2 = 0.968), well inside the 0.5 lp/mm target (Table IV, Fig. 7). On the same 60 test settings, the seed ensemble (MAE 0.115) came close to the GP (0.106), although with larger outliers (RMSE 0.223 against 0.164). The largest error, for both models, is a setting with v = 0.085 mm whose MTF50 (5.74 lp/mm) lies just below the voxel Nyquist frequency (5.9 lp/mm), where sampling dominates. Error grew with resolution — 0.071, 0.124 and 0.274 lp/mm for low, mid and high MTF50 — and was similar for bar, square-edge and round-edge patches (0.148–0.185). The absolute error correlated moderately with the image noise level (Spearman 0.37). Inference took 0.1 ms per patch.

TABLE IV
CNN RESULTS ON THE TEST SPLIT

Quantity	Value
Per-patch MAE / R^2	0.162 ± 0.010 / 0.955
Per-setting MAE / R^2	0.121 ± 0.009 / 0.968
Seed ensemble vs GP (same 60 settings), MAE	0.115 vs 0.106
MAE, MTF50 < 3.1 / 3.1–4.1 / > 4.1	0.071 / 0.124 / 0.274
MAE, bars / square edge / round edge	0.148 / 0.165 / 0.185
Spearman(|error|, noise)	0.37
Generic phantoms: MAE / R^2 / bias	0.603 / 0.559 / +0.143

Errors in lp/mm; ± is the s.d. over three seeds.

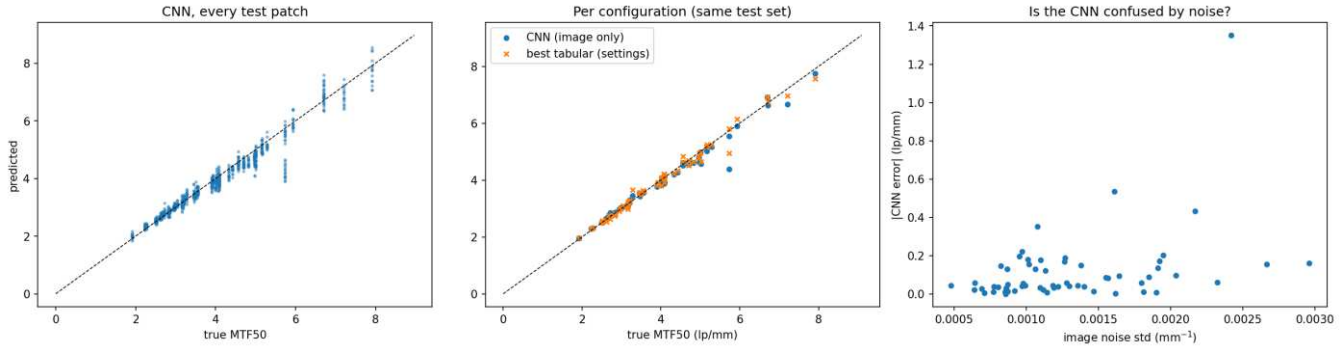


Fig. 7. CNN estimator on the test split. Left: every patch. Centre: mean of 15 patches per setting (circles) with the GP surrogate on the same settings (crosses). Right: per-setting absolute error against image noise level.

H. Local-resolution maps on generic phantoms

On the 20 random-ellipse phantoms the median of each map agreed only moderately with the true MTF50 (MAE 0.603 lp/mm, $R^2 = 0.559$, bias +0.14). Fig. 8 shows why. Inside the object the maps are close to the truth — about 2 lp/mm in the interior of a phantom whose true value is 2.28 — but windows that include the object–air boundary are predicted near 5 lp/mm, and dark (negative-contrast) inserts are predicted too low. Neither air nor negative contrast ever appeared in the training patches, so these are out-of-distribution inputs. Because blur in our simulation is the same everywhere, a perfect map would be uniform; the maps therefore test consistency, not true spatial variation.

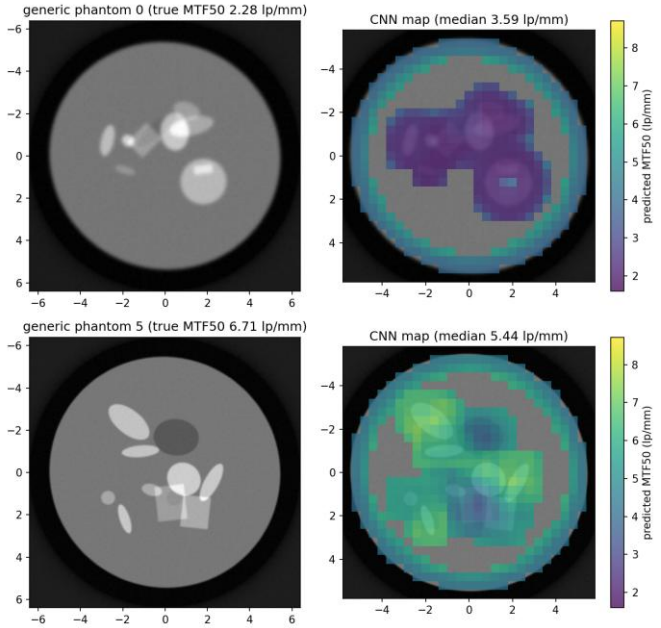


Fig. 8. Generic phantoms (left) and CNN local-resolution maps (right) for two settings. Interior structures are estimated well; the object–air boundary (rim) and the dark negative-contrast ellipse are not represented in the training data and are mis-estimated.

VI. Discussion

A. Was the problem solved?

Within the simulated domain, yes. The surrogate predicts MTF50 to about 0.1 lp/mm — roughly 3 % of a typical value — in about a millisecond, which makes exhaustive exploration of settings practical; the formula alone is about twelve times less accurate. The image estimator achieves nearly the same accuracy without knowing the settings, but only for images resembling its training phantom.

B. Why do physics features help extrapolation?

The derived features rescale the settings into the blur widths the object actually experiences. A magnification or focal spot outside the training range can still produce blur widths inside the range already seen, and the formula feature supplies the correct monotone trend, so the model only has to learn a correction. This is a strong argument for combining physics with learning. It is not a guarantee: voxel-size extrapolation, which involves sampling effects the formula does not describe, degraded for every model.

C. Why did focal spot rank last?

The project brief expected magnification and focal spot to dominate. Our ranking reflects our design space: the focal spot was sampled on a log scale, so most settings have small spots, and its blur is scaled by $(M - 1)/M$, which is small at low magnification. Fig. 5 shows that focal spot becomes decisive at large spots. Sensitivity rankings are therefore statements about a range of settings, not universal constants.

D. Is the CNN measuring resolution?

Partly. Its accuracy on the bar phantom is high, but three shortcuts are possible: noise level, which is higher at small voxel sizes where MTF50 is also high (Spearman 0.37); interpolation texture left by resampling coarse voxels to the common grid; and the fixed contrast of the training inserts. The drop on generic phantoms indicates some reliance on phantom-specific cues. Training with air regions, varied and negative contrasts, and variable noise is the obvious remedy.

E. Ethics and bias

No patient or personal data were used. The main risks are over-trust and hidden bias. All models inherit the assumptions of the simulator, and the dataset contains few high-resolution settings, so errors are largest exactly where users may want the finest detail. A fast, confident number could hide a defect in safety-critical inspection or affect a medical protocol if applied to a real scanner without validation. We therefore present the tools for planning and education, not for certifying image quality, and release all code and data so that the assumptions can be checked.

VII. Limitations

The study is entirely simulated, and no real scanner was used. The simulation is a 2-D central slice, so cone-beam effects

away from the mid-plane are absent. Focal-spot and detector blur are Gaussian and shift-invariant, which is why the local maps cannot be validated against a true spatial variation. Scatter, beam hardening, motion, detector lag and varying noise levels were not modelled, and only one material contrast was used for the edge. The 60-setting test split gives confidence intervals of roughly ±0.03 lp/mm. Finally, the speed comparison is against an inexpensive 2-D simulation; a full 3-D cone-beam simulation typically takes minutes per setting, so the practical speed-up would be larger.

VIII. Conclusion

A Gaussian-process surrogate trained on 280 physics-simulated settings predicts CT MTF50 with 0.106 lp/mm error in about a millisecond, more than ten times more accurately than the classical quadrature formula, and physics-informed features make such surrogates far more robust outside their training range. An image-only CNN matches this accuracy on the training phantom but not on unseen objects. Future work should validate on scans of a physical slanted-edge phantom, move to 3-D cone-beam simulation with spatially varying blur (e.g. ASTRA or TIGRE), broaden the training patches to include air, varied contrast and noise, and use the GP's predictive variance to flag uncertain settings.

Acknowledgment

Use of AI tools: an AI assistant (Claude, Anthropic) was used to help write the simulation and analysis code and to draft parts of this manuscript. The authors reviewed and ran all code, checked every reported number against the generated results, and takes full responsibility for the content.

Data and Code Availability

Code, the simulated dataset (training, validation and testing splits with CSV parameter tables, NPY images and MTF curves) and the trained models are available at https://drive.google.com/drive/folders/1WBMjsRps4XwbloF4X-m204amKYY-hiCs?usp=drive_link.

Video: https://youtu.be/8ytDse_oyiw

References

[1] E. Samei, M. J. Flynn, and D. A. Reimann, "A method for measuring the presampled MTF of digital radiographic systems using an edge test device," *Med. Phys.*, vol. 25, no. 1, pp. 102–113, 1998, doi: 10.1118/1.598165.

[2] S. N. Friedman, G. S. K. Fung, J. H. Siewerdsen, and B. M. W. Tsui, "A simple approach to measure computed tomography (CT) modulation transfer function (MTF) and noise-power spectrum (NPS) using the American College of Radiology (ACR) accreditation phantom," *Med. Phys.*, vol. 40, no. 5, Art. no. 051907, 2013, doi: 10.1118/1.4800795.

[3] C. Tominaga, H. Azumi, M. Goto, M. Taura, N. Homma, and I. Mori, "Tilted-wire method for measuring resolution properties of CT images under extremely low-contrast and high-noise conditions," *Radiol. Phys. Technol.*, 2018, doi: 10.1007/s12194-018-0443-8.

[4] S. Tilley II, J. H. Siewerdsen, and J. W. Stayman, "Model-based iterative reconstruction for flat-panel cone-beam CT with focal spot blur, detector blur, and correlated noise," 2017, arXiv:1706.05418.

[5] M. Bauer, V. Volchkov, M. Hirsch, and B. Schölkopf, "Automatic estimation of modulation transfer functions," 2018, arXiv:1805.01872.

[6] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York, NY, USA: IEEE Press, 1988.

[7] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone-beam algorithm," *J. Opt. Soc. Am. A*, vol. 1, no. 6, pp. 612–619, 1984, doi: 10.1364/JOSAA.1.000612.

[8] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.