



Differentiable Gamma Index-Based Loss Functions: Accelerating Monte-Carlo Radiotherapy Dose Simulation

Sonia Martinot^{1,2,3(✉)}, Nikos Komodakis⁴, Maria Vakalopoulou², Norbert Bus¹, Charlotte Robert³, Eric Deutsch³, and Nikos Paragios¹

¹ Therapanacea, Paris, France

sonia.martinot@hotmail.fr

² CentraleSupélec, Gif-sur-Yvette, France

³ Institut Gustave Roussy, Villejuif, France

⁴ University of Heraklion, Crete, Heraklion, Greece

Abstract. The Gamma index Passing Rate (GPR) is considered the preferred metric to evaluate dose distributions in order to deliver safe radiotherapy treatments. For this reason, in the context of accelerating Monte-Carlo dose simulations using deep neural networks, the GPR remains the default clinical metric used to validate the predictions of the models. However, the optimization criterion that is used for training these neural networks is based on loss functions that are different than GPR. To address this important issue, in this work we introduce a new class of GPR-based loss functions for deep learning. These functions allow us to successfully train neural networks that can directly yield the best dose predictions from a clinical standpoint. Our approach overcomes the mathematical non-differentiability of the GPR, thus allowing a successful application of gradient descent. Moreover, it brings the GPR computation time down to milliseconds, therefore enabling fast trainings. We demonstrate that models trained with our GPR-based loss functions outperform models trained with other commonly used loss functions with respect to several metrics and display a 15% improvement of the GPR over the test data. Code is available at <https://rb.gy/vf5jwv>.

Keywords: Deep Learning · Gamma index · Monte-Carlo · Radiotherapy

1 Introduction

In photon radiation therapy, accurate dose modeling is paramount to ensure treatment plans safely target the tumour. Existing algorithms such as Collapsed Cone Convolution algorithm [1] or Pencil Beam [11] fail to match the Monte-Carlo (MC) radiation transport calculations in terms of precision of the deposited dose [4, 6]. Yet, MC generation of radiotherapy dose distributions remains too time-consuming for clinical adoption. Recent deep learning accelerated MC dose

calculation methods [12, 16] offer a solution for this problem, utilizing common computer vision loss functions. Even if such methods provide a good trade-off between time and performance, training on such loss functions amounts to solving a proxy problem, with no strict assurance to conjointly optimize the clinical validation of the generated dose, which is performed using the Gamma index Passing Rate (GPR).

The GPR is one of the most essential and commonly used clinical evaluation metric for verification of complex radiotherapy dose delivery such as Intensity Modulated Radiation Therapy or Volumetric Modulated Arc Therapy (VMAT) [13]. As such, the GPR provides a clinical criterion to assess the quality of the model’s predictions. Therefore, training directly with the GPR as primary objective would yield more accurate training from a clinical standpoint. However, the GPR has two main limitations that deter from using it as loss function. First, training neural networks in a supervised setting requires a differentiable loss function to allow backpropagation. Yet, the GPR is non-differentiable, thus jeopardizing gradient descent. Secondly, despite efforts to bridge the gap, current Gamma index and GPR computations remain time-consuming, especially when comparing high dimensional dose distributions.

By taking a medical imaging perspective, we circumvent these challenges to incorporate the GPR as an optimization criterion during training of neural networks. According to our knowledge, this is the first study to create a new class of loss functions based on the GPR and to bring the speed of gamma index computations down to milliseconds, both for 2D and 3D dose distributions. We provide a proof-of-concept showcasing deep learning acceleration of MC dose simulations with models trained to optimize the presented GPR-based loss functions. Finally, we study the behavior of the GPR-based loss functions and benchmark them against the Structural Similarity Index Measure (SSIM), the Mean Absolute Error (MAE), and the Mean Squared Error (MSE). Our code and models will be publicly released.

2 Related Work

Loss Functions: When training a neural network on a task, the choice of the loss function is crucial. Loss functions such as the Dice Loss [15], the Focal Loss [7], or the Structural Similarity Index Measure (SSIM) [17] have revolutionized, respectively, segmentation, object detection, and image processing tasks. Moreover, all loss functions do not yield the same impact on the training and inference, as explained in the study introducing the Multiscale-SSIM [19].

This problem becomes even more evident in the medical field, in which models need to ensure reliable performance. For this reason, integrating mathematical objectives that train the models to optimize clinically relevant properties is of utmost importance for their integration into clinical practice.

In light of these considerations, we overcome the mathematical challenge of the GPR and turn this clinical metric into a viable loss function for our task of accelerating the simulation of MC radiotherapy dose distributions. We

provide a family of GPR-based criteria that are therefore in adequacy with clinical requirements.

Gamma Index: The main challenges of computing the gamma index matrix reside in the pixel-wise computation of gamma index values that can be time-consuming proportionally to the dimensionality of the evaluated dose distribution. Prohibitive calculation time hinders the potential of the GPR as loss function. Many works propose ways to decrease the computation complexity, either by changing the mathematical formalism or accelerating the calculations. In [5], Gu et al. use a geometric method with a GPU-accelerated radial pre-sorting technique to speed up calculations. Chen et al. [3] consider reducing the search distance by using a fast Euclidean distance transform.

In this paper, we present an acceleration approach adequate for deep learning frameworks that significantly reduces the calculation speed and enables fast training with our GPR-based loss functions.

3 Methods

3.1 The Gamma Passing Rate

Gamma Index: Let D_r and D_e be two dose distributions ($\mathbb{R}^k \rightarrow \mathbb{R}$), respectively the reference and the evaluated. In our case, the evaluated dose distribution is the model's prediction. To each of them corresponds a grid of points in which each point, P_r of D_r , and P_e of D_e has a coordinate vector, respectively $\vec{d}(P_r)$ and $\vec{d}(P_e)$, and a dose value, $D_r(P_r)$ and $D_e(P_e)$.

Let us consider a point P_r in D_r and the points P_e in a vicinity $V(P_r)$ around P_r . Then the gamma index Γ is defined as a function of real values such that for all $P_r \in D_r$, $\Gamma(P_r)$ writes as follows:

$$\Gamma(P_r) = \min_{P_e \in V(P_r)} \sqrt{\frac{\|\vec{d}(P_e) - \vec{d}(P_r)\|^2}{DTA^2} + \frac{(D_e(P_e) - D_r(P_r))^2}{\Delta^2}} \quad (1)$$

where DTA is the tolerance on the Distance-To-Agreement (DTA), commonly in mm , and Δ is the tolerance on the relative dose difference expressed as a percentage of the reference dose value $D_r(P_r)$. This definition entails that each point P_r has its own gamma index value in Γ , which indicates how close neighbouring points P_e are, both spatially and dose-wise.

GPR: Let us introduce a dose threshold δ and consider a point P_r of the reference distribution such that $D_r(P_r) \geq \delta$. Then, given a DTA and dose tolerance Δ , the evaluated distribution matches the reference at P_r , if the passing criterion is satisfied, i.e. if:

$$\text{Passing criterion: } \Gamma(P_r) \leq 1 \quad (2)$$

The GPR is defined as the percentage of points P_r that satisfy the condition in Eq. 2 while $D_r(P_r) \geq \delta$.

Let $\mathbb{1}_{D_r \geq \delta}$ and $\mathbb{1}_{\Gamma \leq 1}$ be the indicator functions defined such that:

$$\mathbb{1}_{D_r \geq \delta}(P_r) = \begin{cases} 1 & \text{if } D_r(P_r) \geq \delta. \\ 0 & \text{otherwise.} \end{cases} \quad \mathbb{1}_{\Gamma \leq 1}(P_r) = \begin{cases} 1 & \text{if } \Gamma(P_r) \leq 1. \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Then we can write the GPR as follows:

$$GPR(D_r, D_e) = \frac{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r) \cdot \mathbb{1}_{\Gamma \leq 1}(P_r)}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \quad (4)$$

Minimization Problem: With the GPR formulation in Eq. 4, maximizing the GPR amounts to minimizing the corresponding loss function L_{GPR}^δ which draws values in $[0, 1]$:

$$L_{GPR}^\delta(D_r, D_e) = 1 - \frac{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r) \cdot \mathbb{1}_{\Gamma \leq 1}(P_r)}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} = 1 - GPR \quad (5)$$

Due to the fact that the indicator function $\mathbb{1}_{D_r \geq \delta}$ does not depend on Γ , the gradient of L_{GPR}^δ (with respect to the trainable parameters) can be written as follows:

$$\frac{\partial L_{GPR}^\delta}{\partial w} = \frac{1}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \cdot \sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r) \frac{\partial \mathbb{1}_{\Gamma \leq 1}(P_r)}{\partial \Gamma} \frac{\partial \Gamma(P_r)}{\partial w}, \quad (6)$$

where w represents any of the trainable parameters of the neural network.

The problem with the above definition of L_{GPR}^δ is that it generates zero gradients, which is a direct consequence of the fact that the indicator function $\mathbb{1}_{\Gamma \leq 1}(\cdot)$ is stepwise constant with respect to Γ , preventing SGD training. To address this issue, in the following we propose the use of a soft approximation of the objective function L_{GPR}^δ with non-zero gradients.

3.2 Soft Counting with Sigmoid-GPR

To avoid the propagation of null gradients, we propose to use the sigmoid function, $\sigma(x) = (1 + \exp^{-\beta x})^{-1}$, to approximate counting passing voxels. The slope of the sigmoid depends on the value of its sharpness β that we consider as a hyperparameter.

Moreover, we note that for all $P_r \in D_r$, it stands that:

$$\lim_{\beta \rightarrow +\infty} \sigma(\beta \cdot (1 - \Gamma(P_r))) = \mathbb{1}_{\Gamma \leq 1}(P_r) \quad (7)$$

Hence, the asymptotic behaviour of the sigmoid function combined with shifting the gamma index values can provide an estimate of the count of passing voxels by summation over all points P_r . The accuracy of the estimation then depends on the value of β : the bigger the β , the more precise the estimation will be.

Thus, we approximate the loss L_{GPR}^δ in Eq. 5 with $L_{\sigma-GPR}^\delta$ defined using the sigmoid function:

$$L_{\sigma-GPR}^\delta = 1 - \frac{\sum_{P_r \in D_r} \sigma(\beta \cdot (1 - \Gamma(P_r))) \mathbb{1}_{D_r \geq \delta}(P_r)}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \quad (8)$$

$L_{\sigma-GPR}^\delta$ is differentiable everywhere and, provided the sharpness β is not too high, gradients are non-zero and allow gradient descent to update the model's weights during backpropagation.

Given Eq. 7, we remark that $L_{\sigma-GPR}^\delta$ accurately approximates the true GPR loss function, i.e., $L_{\sigma-GPR}^\delta \rightarrow L_{GPR}^\delta$ as $\beta \rightarrow +\infty$.

Annealing Schedule of β : In light of the equations above, we propose to consider β as a hyperparameter. At the beginning of training, the model usually predicts poorly and the majority of voxels fail to satisfy the gamma index passing criterion. This implies that the corresponding loss computed with $L_{\sigma-GPR}^\delta$ generates zero gradients everywhere if the value of β is set too high. To avoid this behaviour, we propose an annealing schedule for β that starts with low initial values and progressively increases β over the training. Moreover, when $\beta \sim 0^+$, the Taylor series expansion of the sigmoid function yields:

$$\sigma(\beta \cdot (1 - \Gamma(P_r))) \sim \beta \cdot (1 - \Gamma(P_r)) \quad (9)$$

Given Eq. 9, we can write the Taylor expansion of $L_{\sigma-GPR}^\delta$ when $\beta \sim 0^+$:

$$L_{\sigma-GPR}^\delta \sim 1 - \beta + \frac{\sum_{P_r \in D_r} \Gamma(P_r) \mathbb{1}_{D_r \geq \delta}(P_r)}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \quad (10)$$

Consequently, we introduce the loss function $L_\Gamma^\delta(D_r, D_e)$ to model the linear behaviour of $L_{\sigma-GPR}^\delta$ at the start of the annealing schedule:

$$L_\Gamma^\delta(D_r, D_e) = \frac{\sum_{P_r \in D_r} \Gamma(P_r) \cdot \mathbb{1}_{D_r \geq \delta}(P_r)}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \quad (11)$$

As the training continues and the loss decreases, the annealing scheme proceeds in progressively increasing β in order to improve the approximation of the GPR loss L_{GPR}^δ defined in Eq. 5. As β increases and is acquiring larger values, minimizing the loss amounts to getting failing voxels (voxels with $\Gamma > 1$) to satisfy the passing criterion.

To model the behaviour of $L_{\sigma-GPR}^\delta$ at this stage (i.e. as β is acquiring larger values), we modify L_Γ^δ in Eq. 11 by introducing the loss function $L_{\Gamma>1}^\delta$. To prevent backpropagation of zero gradients with respect to $\mathbb{1}_{\Gamma>1}$, we use the *stopgrad* operation:

$$L_{\Gamma>1}^\delta(D_r, D_e) = \frac{\sum_{P_r \in D_r} \Gamma(P_r) \cdot \mathbb{1}_{D_r \geq \delta}(P_r) \cdot \text{stopgrad}(\mathbb{1}_{\Gamma>1}(P_r))}{\sum_{P_r \in D_r} \mathbb{1}_{D_r \geq \delta}(P_r)} \quad (12)$$

For the sake of characterizing the behaviour of $L_{\sigma-GPR}^\delta$ in Eq. 8, we also study trainings that involve the use of loss functions L_Γ^δ and $L_{\Gamma>1}^\delta$ in the following experiments.

3.3 Accelerating Gamma Index Matrix Computations

Having a differentiable GPR loss does not make it directly applicable for neural network training since, by definition, it requires iterating over all voxels in the given distributions, therefore leading to prohibitive computation time when considering high-resolution distributions. To deal with this issue, we propose an accelerated version of GPR for faster calculations.

To avoid physical incoherence when computing gamma index values, we sample the evaluated and reference distributions to the resolution 1 mm^3 with bilinear interpolation. By the definition in Eq. 1, one can observe that the evaluated voxels located farther than $DTA\text{ mm}$ from P_r automatically yield a gamma index superior to 1. Thus we limit the search within an invariant vicinity defined by the chosen DTA . More precisely, the gamma index value of a reference point then stems from comparing gamma values computed with voxels in a cube comprising $(2 \times DTA + 1)^k$ voxels, in the case of k dimensional dose distributions. We then use unfolding to extract sliding local blocks of the evaluated distribution generated by the model. This operation creates one channel per voxel in the vicinity defined by the DTA . We then apply the minimum operation over the channel dimension to get the minimal gamma index value.

The approach enables fast computation of the gamma index distribution Γ , which is necessary for the calculations of $L_{\sigma-GPR}^{\delta}$, L_{Γ}^{δ} and $L_{\Gamma>1}^{\delta}$ presented in Eq. 8, 11, 12. Computation times are discussed in Sect. 5.

4 Dataset and Experimental Design

Dataset: We carried out the experiments on the publicly available dataset presented in [10] comprising 50 patients treated with VMAT plans. Each patient has a reference dose distribution computed from 1×10^{11} particles and a low precision simulation computed from 1×10^9 particles. The main goal of the methods benchmarked on this task is to generate the high precision simulation of the dose from the available low precision one. More details about the dataset can be found in the original publication. For our experiments, we split patients to 35-5-10 for respectively, the train, validation and test sets. The cases in the dataset correspond to various anatomies and therefore, we split them as equally as possible between sets to avoid biases.

Even though our approach enables training on 3D dose distributions, the dataset comprises a small number of samples. Thus, we decided to carry out the experiments in 2D to favour significant experiments and a relevant benchmark. In this setting, a training sample corresponds to an axial slice of a patient's dose volume. The 2D training dataset therefore comprises around 11k training samples, where a sample is a pair of corresponding slices of low precision and high precision dose simulation.

Preprocessing: We normalized both low precision and reference distributions using the average dose maximum computed over the reference dose volumes

from the training set. We then applied the same normalization on the validation and test sets. To enable batch training, we padded each training sample with zeros in order to match a fixed size of 256×256 . To further help the model generate accurate dose predictions, we added the corresponding CT slice as second input channel to incorporate the corresponding anatomy. We applied minimax normalization to CT volumes so voxel values remain in $[0, 1]$ range.

Model: In all experiments, the model is a standard UNet architecture [14] with skip connections between the encoder and the decoder. The encoder part of the model performs downsampling twice with convolutional layers using 4×4 filters and a stride of 2. Symmetrically, transposed convolutions upsample feature maps in the decoder. Each stage of the UNet comprises two convolution blocks before downsampling or upsampling. Much like the convolutional block presented by Liu et al. in [8], a convolution block first applies a convolution with 7×7 filters and 3×3 padding, and then two convolutions with 3×3 filters to further process the features maps. Each convolution is followed by Gaussian Error Linear Units (GELU) activation units. The block ends with a residual connection to keep high frequency details from the block’s input. Overall, the model has around 10 million trainable parameters.

Optimization Set-Up: In all trainings, we trained the model using AdamW optimizer [9]. We set the initial learning rate to $3e^{-4}$ and decreased it progressively during the training when the validation loss stagnated. Weight decay was set to $5e^{-4}$ and batch size to 16. We trained for 20k iterations on a NVidia GeForce RTX 3090 GPU. The trainings were stopped when overfitting appeared by adopting the early stopping strategy. With this training scheme, early stopping occurred after around 15k iterations, when the validation loss fails to decrease 2% after 500 iterations.

Loss Functions: To train with the GPR-based loss using sigmoid count $L_{\sigma-GPR}^{\delta}$ presented in Eq. 8, we designed the following annealing schedule for the sharpness parameter β . We set the initial value of β to 2×10^{-2} for the first 150 iterations. Then, β increased by a factor of 5% every 50 iterations until it reached an intermediate value of 3 where updates slowed down to 5% every 100 iterations. Increasing updates stopped when β reached a chosen ceiling value of $\beta_{max} = 5$. Setting β_{max} prevented the slope of the sigmoid from getting too sharp and the loss from encountering a vanishing gradient problem, which would stop the updates of gradient descent. Additional benchmarks with the approximating loss functions L_{Γ}^{δ} and $L_{\Gamma>1}^{\delta}$ have been also conducted, in order to better characterize the behaviour of $L_{\sigma-GPR}^{\delta}$ for small values of β .

For all GPR-based functions, we set the dose threshold δ to 20%. This means that, while loss functions compute the gamma index distribution by considering all voxels, the computed approximated GPR value takes into account only voxels P_r for which the dose value is superior to 20% of the maximum dose of the reference distribution, i.e. $D_r(P_r) \geq 20\% \cdot \max_{P_r \in D_r} D_r(P_r)$.

Table 1. Evaluation metrics over the dose distributions comprised in the test set. Different benchmarks over the considered loss functions for different metrics are highlighted with their mean and standard deviation. With bold we indicate the best performing methods per metric.

Loss function	GPR 2%/2mm	GPR 3%/2mm	GPR 3%/3mm	SSIM (%)	MAE	MSE
MAE	47.6 ± 7.6	53.8 ± 8.4	66.5 ± 10.0	83.3 ± 9.2	0.39 ± 0.10	0.51 ± 0.28
MSE	50.2 ± 7.8	56.7 ± 8.7	69.5 ± 10.3	83.9 ± 11.5	0.34 ± 0.09	0.43 ± 0.29
SSIM+MAE	46.3 ± 8.8	52.3 ± 9.8	64.6 ± 11.9	94.0 ± 8.8	0.35 ± 0.09	0.74 ± 0.36
SSIM+MSE	49.1 ± 10.0	55.3 ± 11.1	67.5 ± 13.2	93.3 ± 3.1	0.30 ± 0.07	0.30 ± 0.15
L_T^δ	57.7 ± 3.3	65.0 ± 3.5	79.2 ± 4.3	87.5 ± 7.6	0.28 ± 0.07	0.30 ± 0.23
$L_{T>1}^\delta$	57.2 ± 3.9	64.6 ± 4.1	79.0 ± 4.4	86.7 ± 8.1	0.27 ± 0.07	0.25 ± 0.16
$L_{\delta-GPR}^\delta$	59.3 ± 3.3	66.8 ± 3.6	81.4 ± 4.0	88.2 ± 7.5	0.24 ± 0.06	0.22 ± 0.16

To benchmark against our proposed GPR-based loss functions, we considered several other loss functions commonly used in computer vision. The benchmark includes the MAE and the MSE for a comparison with pixel-wise errors. Finally, we considered the combination of pixel-wise errors with the SSIM. More precisely, the benchmark includes SSIM-MAE and SSIM-MSE, which are the equally weighted sum of respectively the SSIM and MAE, and the SSIM and MSE. For each training on the loss functions considered above, we used the exact same model architecture and optimization strategy, in order to promote the reliability and fairness of the comparison.

5 Results

5.1 Training with GPR-Based Loss Functions

Extensive quantitative comparison on the test set for each training, using the MAE, MSE, SSIM and GPR with various values of DTA and dose tolerance δ are summarised in Table 1. As the test set comprises 10 patients, we computed the metrics over each slice of each patient’s volume, and then average over the test set per considered metric. Results point out that models trained with GPR-based loss functions tend to outperform others with respect to the GPR, the MAE and MSE. In contrast, models trained with SSIM-MAE and SSIM-MSE show the highest SSIM scores. With a closer look however, one can observe that they report among the lowest performance for the rest of the metrics. This result indicates that the SSIM may not be a well-suited metric to evaluate the quality of dose distributions, since it seems to be biased.

To assert statistical significance of the results, we take an in-depth look at each patient in the test set to explain the high standard deviation values observed in Table 1. Boxplots a), b) and d) in Fig. 1 point out the presence of an outlier patient case on which models tend to fail with respect to the GPR, SSIM and MSE. In contrast with SSIM, MSE and MAE-trained models, we observe that models trained with GPR-based loss functions not only display robustness to this outlier, but also show smaller standard deviation over the whole test set.

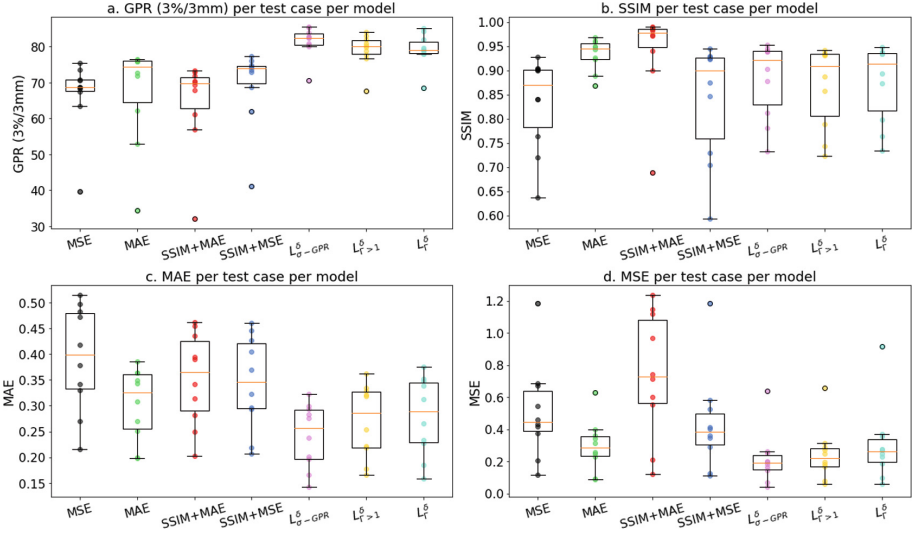


Fig. 1. Boxplots representing the evaluation metrics achieved by trained models for each case in the test set depending on the loss function used for training. The y axis indicates the values of the considered metric. The x axis specifies the loss function with which the corresponding model was trained.

Figure 1 also allows to compare discrepancies within the family of GPR-based loss functions. While all of them produce better performing models with respect to all evaluation metrics except the SSIM, the loss function with sigmoid counting $L_{\sigma-GPR}^{\delta}$ outperforms L_{Γ}^{δ} and $L_{\Gamma>1}^{\delta}$. We explain this behaviour by the fact that both L_{Γ}^{δ} and $L_{\Gamma>1}^{\delta}$ focus only on minimizing gamma index values, and not directly maximizing the number of voxels satisfying the passing criterion. We conclude that $L_{\sigma-GPR}^{\delta}$ yields better maximization of the GPR and is therefore the better approximation of the true GPR loss function L_{GPR}^{δ} .

The MSE-trained model outperforms other models trained with non GPR-based loss functions with respect to the GPR, so we chose to display its dose prediction conjointly with the dose generated by the $L_{\sigma-GPR}^{\delta}$ trained model in Fig. 2. Although both trainings achieved convergence, the prediction of the MSE-trained model manifests important artefacts at the bottom of the generated dose. Additionally, the dose itself seems to be smoother than the dose predicted with the $L_{\sigma-GPR}^{\delta}$ training. Finally, the MSE-trained model appears to overestimate the dose in low-dose regions to a greater extent than the $L_{\sigma-GPR}^{\delta}$ -trained model.

5.2 Speed-Up of GPR-Acceleration Approach

In an effort to promote the GPR-based loss functions as viable deep learning optimization criteria that allow fast error computations and training, we had to accelerate gamma index computations. To quantify the extent of our accel-

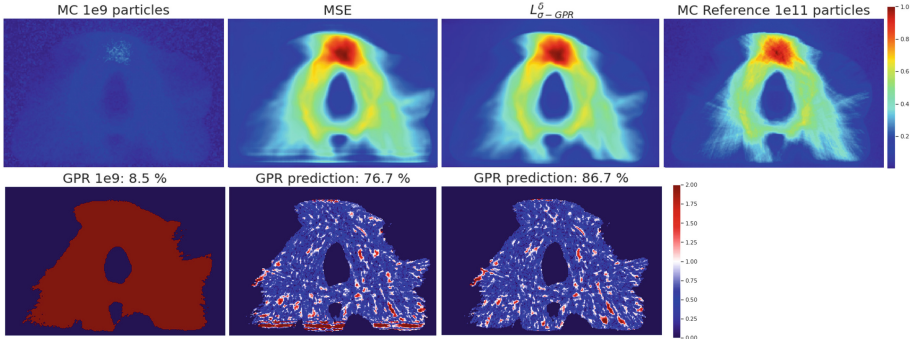


Fig. 2. First row from left to right: a single slice of the 1e9 dose volume, predictions of models trained with MSE and $L^{\delta}_{\sigma-GPR}$, and reference 1e11 dose. Second row: gamma index maps for the three different representations.

Table 2. Speed comparison of metrics computed over 2D or 3D dose distributions.

Metric	3D dose Time(ms)	2D dose Time(ms)
MSE	0.14 ± 0.02	0.13 ± 0.03
MAE	0.23 ± 0.06	0.19 ± 0.04
SSIM	27.49 ± 7.73	5.06 ± 3.32
$L^{\delta}_{\sigma-GPR}$, $L^{\delta}_{T>1}$, L^{δ}_{T}	30.54 ± 0.01	4.51 ± 0.00
Exhaustive	985 ± 515	8.01 ± 2.28
PyMedPhys	> 1 second	> 100 ms

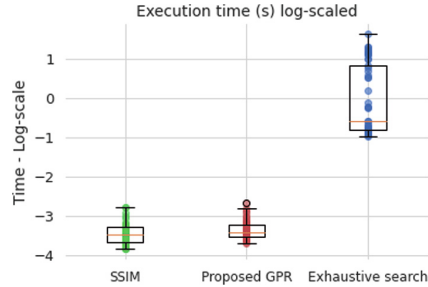


Fig. 3. Boxplots of execution times of the SSIM, our proposed approach and the exhaustive search method on 3D dose distributions.

eration approach, we benchmark against two methods. The first one is a GPU-accelerated exhaustive search approach in a limited vicinity of 3 mm^3 around the considered reference voxel. The second is an open-source tool from PyMedPhys [2] which makes use of acceleration ideas from Wendling et al. [18] and executes on CPU and is single-threaded. Regarding the latter, we limit the interpolation ratio to 2 to have a fair comparison.

The time estimation was twofold. We timed each evaluation metric and GPR-based loss functions on 3D or 2D distributions stemming from the MC dataset used for the experiments. 3D dose distributions were interpolated to resolution 1 mm^3 and of shape $128 \times 200 \times 200$, comprising around 5×10^6 voxels. The 2D dose distributions comprised axial slices of the 3D dose distributions and were interpolated to a size 400×400 . For the GPR calculations, we set the DTA and Δ to respectively 2 mm and 2%. Execution times are displayed in Table 2.

Figure 3 and Table 2 highlight that our approach has equivalent speed to that of the SSIM. Compared to the exhaustive search method, our approach improves

the speed of gamma index computations by a factor of at least 30 in the case of 3D dose distributions and twofold for 2D distributions. Consistently with these results, we note that trainings took around 24 h for SSIM-MAE, SSIM-MSE and GPR-based loss function, whereas they lasted for 15 h for experiments with the MAE and the MSE. Results therefore validate our GPR-based loss functions to efficiently train deep neural networks. Nonetheless, our comparison is limited to speed assessment and does not encompass RAM usage and precision considerations. Although our approach highlights significant speed gain in the computation of the GPR metric and, by extension, of the GPR-based loss functions presented in this study, it comes at the price of an increased RAM usage caused by the unfolding operation.

We make the remark that for all loss functions, the obtained GPRs do not meet the 95% GPR threshold indicating clinical validation. Nevertheless, the goal of the experiments was to show the benefits of optimizing directly the clinical metric during training and results support that statement.

6 Conclusion

Adopting the correct optimization criterion is essential to train deep learning models adequately with the task they are designed to solve. For the task of accelerating MC radiotherapy dose simulation with deep learning, this work proves that directly optimizing models with the clinical validation metric yields significant improvement in predicted dose quality when compared to other loss functions. We provide a fast computation of the GPR to enable such results. Moreover, the GPR is a similarity metric for distributions in general, and may be applied to other tasks such as radiotherapy dose generation or even finding adversarial examples for generative adversarial networks. Future work will focus on addressing the remaining limitations of our approach and assessing the potential of our new class of loss functions in solving other deep learning tasks.

References

1. Ahnesjö, A.: Collapsed cone convolution of radiant energy for photon dose calculation in heterogeneous media. *Med. Phys.* **16**(4), 577–592 (1989)
2. Biggs, S., et al.: Pymedphys: a community effort to develop an open, python-based standard library for medical physics applications. *J. Open Source Softw.* **7**(78), 4555 (2022)
3. Chen, M., Lu, W., Chen, Q., Ruchala, K., Olivera, G.: Efficient gamma index calculation using fast Euclidean distance transform. *Phys. Med. Biol.* **54**(7), 2037 (2009)
4. Deng, Y., et al.: Comparison of pencil beam and Monte Carlo calculations with ion chamber array measurements for patient-specific quality assurance. *Radiat. Med. Prot.* **3**(3), 115–122 (2022)
5. Gu, X., Jia, X., Jiang, S.B.: GPU-based fast gamma index calculation. *Phys. Med. Biol.* **56**(5), 1431 (2011)

6. Lee, B.I., Boss, M., LaRue, S.M., Martin, T.W., Leary, D.: Comparative study of the collapsed cone convolution and monte carlo algorithms for radiation therapy planning of canine sinonasal tumors reveals significant dosimetric differences. *Veterinary Radiol. Ultrasound Official J. Am. Coll. Veterinary Radiol. Int. Veterinary Radiol. Assoc.* **63** (2021)
7. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE ICCV, pp. 2999–3007 (2017)
8. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11976–11986 (2022)
9. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
10. Martinot, S., Bus, N., Vakalopoulou, M., Robert, C., Deutsch, E., Paragios, N.: High-particle simulation of monte-carlo dose distribution with 3D convlstm. In: MICCAI, pp. 499–508 (2021)
11. Mohan, R., Chui, C., Lidofsky, L.: Differential pencil beam dose computation model for photons. *Med. Phys.* **13**(1), 64–73 (1986)
12. Neph, R., Lyu, Q., Huang, Y., Yang, Y.M., Sheng, K.: Deepmc: a deep learning method for efficient monte carlo beamlet dose calculation by predictive denoising in magnetic resonance-guided radiotherapy. *Phys. Med. Biol.* **66**(3), 035022 (2021)
13. Quan, E., et al.: A comprehensive comparison of IMRT and VMAT plan quality for prostate cancer treatment. *Int. J. Radiat. Oncol. Biol. Phys.* **83**, 1169–78 (2012)
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: MICCAI, pp. 234–241 (2015)
15. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 240–248 (2017)
16. Vasudevan, V., Huang, C., Simiele, E., Yu, L., Xing, L., Schuler, E.: Combining monte carlo with deep learning: Predicting high-resolution, low-noise dose distributions using a generative adversarial network for fast and precise monte carlo simulations. *Int. J. Radiat. Oncol. Biol. Phys.* **108**(3), S44–S45 (2020)
17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
18. Wendling, M., et al.: A fast algorithm for gamma evaluation in 3D. *Med. Phys.* **34**(5), 1647–1654 (2007)
19. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imaging* **3**(1), 47–57 (2016)