



Inducing robustness and plausibility in deep learning optical 3D printer models

DANWU CHEN^{1,2,*} AND PHILIPP URBAN^{1,3}

¹Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

²Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany

³Norwegian University of Science and Technology NTNU, Gjøvik, Norway

*danwu.chen@igd.fraunhofer.de

Abstract: Optical 3D printer models characterize multimaterial 3D printers by predicting optical or visual quantities from material arrangements or tonal values. Their accuracy and robustness to noisy training data are crucial for 3D printed appearance reproduction. In our recent paper [Opt. Express 29, 615 (2021)], we have proposed a pure deep learning (PDL) optical model and a training strategy achieving high accuracy with a moderate number of training samples. Since the PDL model is essentially a black-box without considering any physical grounding, it is sensitive to outliers or noise of the training data and tends to create physically-implausible tonal-to-optical relationships. In this paper, we propose a methodology to narrow down the degrees-of-freedom of deep-learning based optical printer models by inducing physically plausible constraints and smoothness. Our methodology does not need any additional printed samples for training. We use this approach to introduce the robust plausible deep learning (RPDL) optical printer model enhancing robustness to erroneous and noisy training data as well as physical plausibility of the PDL model for selected tonal-to-optical monotonicity relationships. Our experiments on four state-of-the-art multimaterial 3D printers show that the RPDL model not only almost always corrects implausible tonal-to-optical relationships, but also ensures significantly smoother predictions, without sacrificing accuracy. On small training data, it even outperforms the PDL model in accuracy by up to 8% indicating a better generalization ability.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

Optical printer models are functions that predict a print's optical properties given the arrangement or ratio of printing materials. They are the prerequisite to accurately reproduce color [1–3], translucency [4,5] or joint color and translucency [6] in multi-material 3D printing employing materials colored in Cyan (C), Magenta (M), Yellow (Y), Black (K), White (W), and a fully Transparent (T) material.

Proposed optical printer models can be classified into phenomenological models [7–22], models based on the Radiative Transfer Equation (RTE) or its simplifications [23–28] and neural-network-based models [29–32]. We refer to our previous paper [33] for a discussion of these models.

Our previous paper [33] has proposed two deep-learning models to optically characterize multimaterial 3D printing systems: First, the *Pure Deep Learning* (PDL) model that does not rely on any physical grounding; Second, the *Deep-Learning-Linearized Cellular Neugebauer* (DLLCN) model that uses deep learning to multi-dimensionally linearize the tonal-value-space of a cellular Neugebauer model. Both models achieve high accuracy with a moderate number of training prints. Due to larger degrees-of-freedom combined with learning strategies reducing overfitting, the PDL model is more accurate than the DLLCN model w.r.t. spectral, color and translucency errors.

However, a shortcoming of a purely empirical deep-learning-based approach is that it does not consider physical/perceptual knowledge of relationships between material ratios (i.e. tonals) and

the resulting optical/visual properties. This results in implausible tonal-to-optical predictions. One way to ensure physical plausibility is using a deep learning model to adjust the parameters of a physical or partly physical model such as proposed for the DLLCN model. Unfortunately, physical models may lack the degrees-of-freedom to accurately consider all influencing factors, such as complex physical material mixing (e.g. between support and build materials at the object's surface) or post-process treatment, which likely adversely impact their prediction performance.

In this paper, we show how to induce physically-based heuristics into purely empirical models. A plausible heuristic is, for instance, that the print's reflectance factor (or lightness) does not increase with an increasing fraction of black material assuming that the black material has the maximum absorption of all available printing materials. Such monotonicity relationship between material ratios and measurable optical quantities applies also for translucency: Increasing the fraction of transparent material (a material with negligible absorption and scattering) in the material mixture does not decrease the translucency α -value [34] of the resulting print.

Another issue of any optical characterization process is the quality and plausibility of the data to fit or train the model. Printing and measurement errors may cause implausible training data violating the heuristics mentioned above. Figure 1 shows an example where the training data has errors leading to violations in the monotonic relationship between lightness and black material usage. Printer variability (spatial, temporal) also induces noise into the training data, aggravating the challenge of noisy predictions from optical printer models. Figure 2 shows an example where a purely empirical PDL model [33] has bumpy and implausible predictions. For reproducing a distinct optical quantity, an optical printer model needs to be inverted using constraint optimization to obtain the corresponding material ratios or tonals [35–38] – this process is called *separation*. Due to local non-monotonicity caused by noisy training data, such inversions will cause banding artifacts in gradients [39] reducing the print quality [40]. A relatively simple solution to reduce noise and outliers in the training data is printing and measuring the same sample multiple times and only consider the median for training. This reduces data noise but increases data collection efforts by a multiple.



Fig. 1. Lightness L^* as a function of black material K with other materials fixed on a tonal case of the *Mimaki 2* dataset (the fixed tonal values are shown at the lower left of the figure), and the resulting colors. The L^* values are shown as white numerical text, with italic font indicating L^* -vs- K monotonicity violations. The PDL/RPDL prediction in this figure is from a single PDL/RPDL model respectively. The figure shows that the training data itself has errors leading to monotonicity violations, and that the PDL model overfits to the erroneous data without considering monotonicity. In contrast, the proposed RPDL model ensures monotonicity despite the data outlier, indicating better robustness against errors in training data.

Figure 3 shows a smooth C,M,Y,K separation example, which would result in an artifact-free physical printout. In contrast, the PDL prediction shows banding artifacts that would not appear in the physical printout. This indicates that a separation (i.e. inversion) based on the PDL model

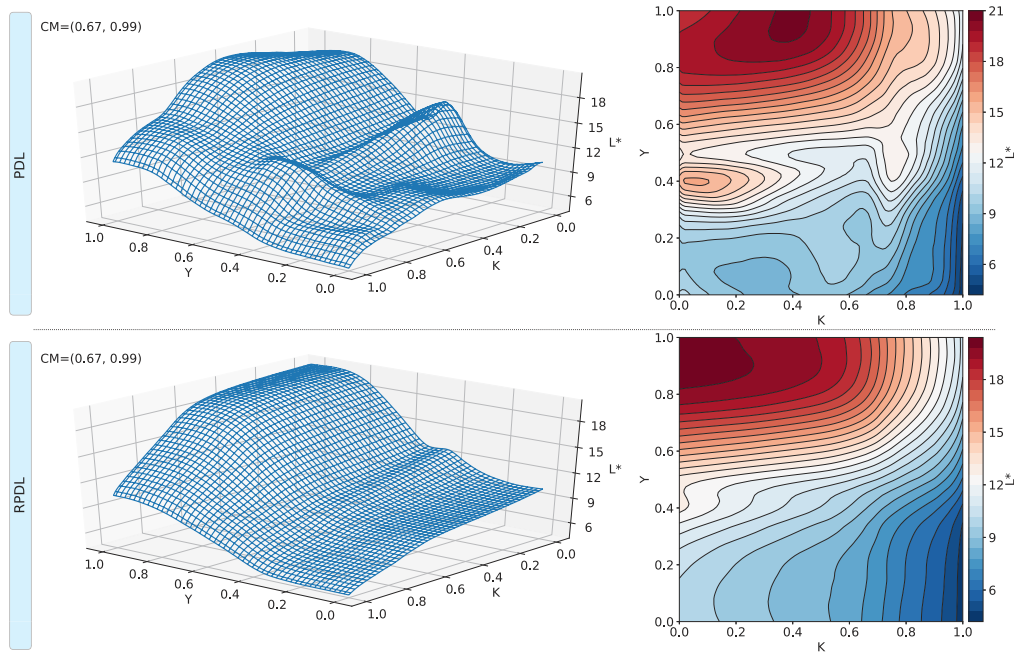


Fig. 2. The 3D plot and contour plot of L^* as a function of Y and K with other materials fixed on a tonal case of the "Mimaki 2" dataset (The fixed values are shown at the upper left corner). The first row corresponds to the PDL model [33] and the second to the proposed RPDL model. Notice the skewed curves and isolated "islands" in the contour plot for the PDL predictions. Drawing a profile across such a contour island along the K direction will result in a bumpy L^* -vs- K curve violating monotonicity. In contrast, the RPDL predictions do not possess such islands and show much smoother contour curves ensuring monotonicity, indicating better robustness and plausibility.

possesses such artifacts as well, which will be reflected in physical printouts and is unacceptable for color critical applications in which texture detail preservation is crucial.

In this paper, we propose a methodology to narrow down the degrees-of-freedom of deep-learning based optical printer models by inducing physically plausible constraints and smoothness. Our methodology does not need any additional printed samples for training. We use this approach to introduce the Robust Plausible Deep Learning (RPDL) optical printer model enhancing robustness to erroneous and noisy training data as well as physical plausibility of the PDL model for selected tonal-to-optical relationships. In particular, we make the following contributions:

1. We introduce a learning strategy to induce monotonicity heuristics into the PDL model by proposing a new derivative-based loss function that is evaluated in the training process by random tonal value re-sampling.
2. We select physical plausible monotonicity relationships between lightness (CIE L^*) and black material as well as between translucency (α -value) and transparent material.
3. To make the PDL model more robust to noisy input data, we induce a smoothness heuristic of the tonal-to-optical relationship by a new second-derivative-based loss function that is evaluated during training by random tonal value re-sampling.

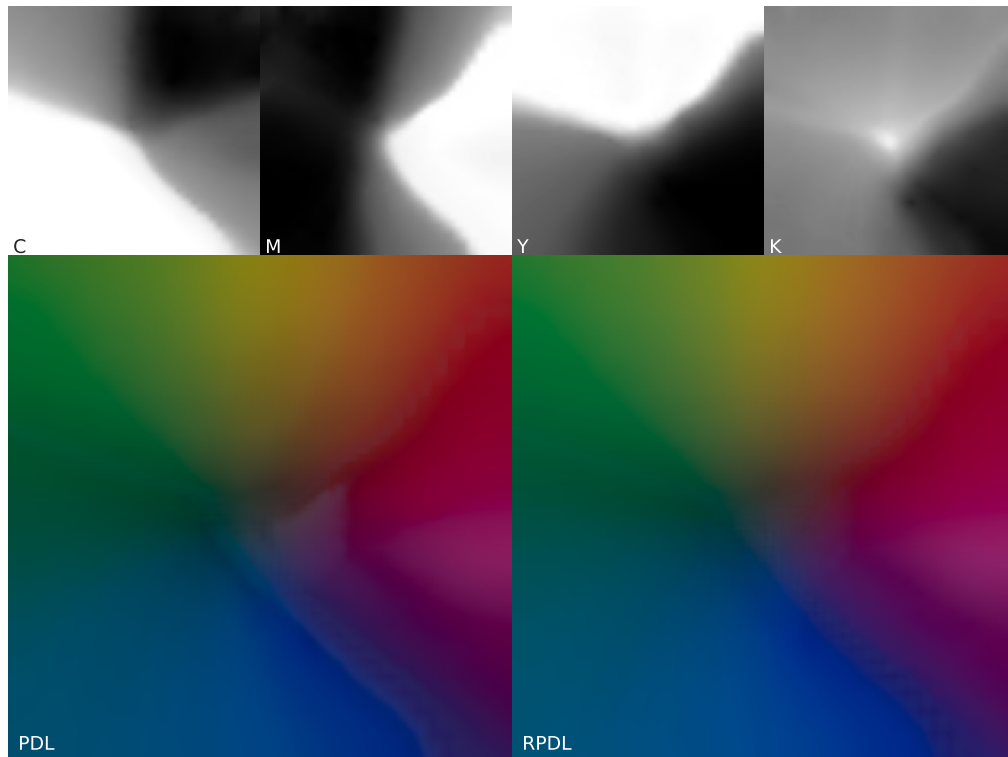


Fig. 3. C,M,Y, and K separations of the gamut mapped CIE- a^*b^* plain for CIE- $L^* = 20$ for the Mimaki 3DUJ-553 3D printer based on the RPD model and the color predictions of the PDL and RPD models.

4. We propose an automatic hyper-parameter optimization strategy to combine the new loss functions and PDL's original loss functions considering an upper threshold for color accuracy losses.

We show on four datasets from state-of-the-art multimaterial 3D printers that the proposed strategy improves model plausibility and robustness without sacrificing accuracy. In our experiments, the RPD models almost always do not show violations of the two induced monotonicity constraints, which is a prerequisite for banding artifact-free separations and as a consequence also artifact-free physical printouts. This is crucial for color critical applications in which preserving texture details is important.

2. Pure deep learning (PDL) optical printer model

The PDL model [33] operates not directly on material ratios but in a tonal space. In the next section we describe the transformation from tonals to material ratios.

2.1. Tonal to material mixture transformation

A 3D printing system with k materials can be controlled by a tonal space of $m = k - 1$ dimensions since material ratios must sum up to one in 3D printing (*unity condition*) and the fraction of one material is implicitly defined by the sum of the other material fractions. In this paper, the material not explicitly connected to tonals is white. We aim to allow all values in the hypercube $[0, 1]^m$ to be valid inputs of the PDL or RPD model. We denote $\mathcal{T} = [0, 1]^m$ to be the tonal space. It does not directly represent material ratios but needs to be converted to material ratios ensuring

the unity condition by a transform $\mathbf{H} : \mathcal{T} \mapsto [0, 1]^m$, with $\|\mathbf{H}(t)\|_1 \leq 1, \forall t \in \mathcal{T}$ from which the fraction of the white material can be computed as follows $t_w(t) = 1 - \|\mathbf{H}(t)\|_1$. \mathbf{H} is part of the 3D printing pipeline before 3D halftoning (see Fig. 4) and is a function concatenation $\mathbf{H} = \mathbf{P} \circ \mathbf{Q}$, where $\mathbf{P}(t) = (t_1/\max\{\|t\|_1, 1\}, \dots, t_m/\max\{\|t\|_1, 1\})$ is a projection of the tonals to ensure the unity condition. $\mathbf{Q} : [0, 1]^m \mapsto [0, 1]^m$ is an invertible transform similar to the transforms from nominal to effective tonals (e.g. 1D-per-tonal curves as described in [41]). This supports the selection of printing patches corresponding to a more uniform distribution of optical quantities to fit/train the PDL model.



Fig. 4. Tonal to print workflow [33].

2.2. Structure of the PDL model

The PDL model is a function $\mathbf{PDL} : \mathcal{T} \mapsto \mathcal{S} \times \mathcal{A}$ predicting spectral reflectances $r \in \mathcal{S} = [0, 1]^N$ and translucency $\alpha \in \mathcal{A} = [0, 1]$ from tonal values $t \in \mathcal{T}$. Here N is the number of considered wavelengths that is set to $N = 31$ assuming a uniform sampling of the visible wavelength range [400nm, 700nm] in 10nm steps.

The PDL model is a multi-path, fully-connected neural network (see Fig. 5) with two paths, one for predicting reflectance and one for translucency. Built upon the tonal input layer is the *trunk* consisting of several hidden layers to learn generic features across tasks (marked as *shared layers* in Fig. 5). It splits into two *branches* one for predicting reflectance/color and one for translucency to learn task-specific features via extra hidden branch-layers followed by the branch's output layer.

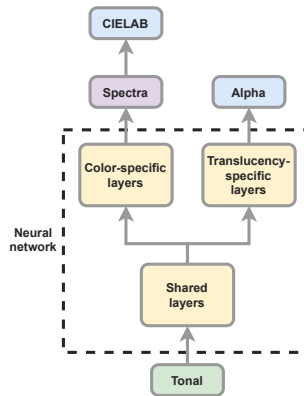


Fig. 5. PDL model's neural network structure [33] is also used by the RPDL model.

Specifically, the trunk has 4 hidden layers each has 200, 1500, 1500 and 1500 neurons respectively. The reflectance branch has one hidden layer with 200 neurons and an output layer with N neurons corresponding to N -dimensional reflectance that is then converted to 3-dimensional CIELAB, and the translucency branch has one hidden layer with 30 neurons and an output layer with 1 neuron corresponding to the α value.

All hidden layers use the *leaky Rectified Linear Unit* (leaky-ReLU) activation function [42] and the two output layers use a variant of sigmoid activation function introduced in the PDL work [33].

2.3. Loss function of the PDL model

The loss function \mathbf{E}_{PDL} of the PDL model consists of three parts: the spectral Root-Mean-Square Error (RMSE) \mathbf{E}_{ref} , the CIEDE2000 error \mathbf{E}_{col} computed for specific viewing conditions (illuminant, observer), and the α -based translucency error \mathbf{E}_{tra} . The final loss function is a weighted average of these three loss functions:

$$\mathbf{E}_{\text{PDL}} = \mathbf{E}_{\text{col}} + a\mathbf{E}_{\text{ref}} + b\mathbf{E}_{\text{tra}}, \quad \text{with} \quad (1)$$

$$\mathbf{E}_{\text{ref}}(r_p, r_m) = \sqrt{\frac{1}{N} \|r_p - r_m\|_2^2} \quad (2)$$

$$\mathbf{E}_{\text{col}}(r_p, r_m) = \Delta E_{00}(\mathbf{LAB}(r_p), \mathbf{LAB}(r_m)) \quad (3)$$

$$\mathbf{E}_{\text{tra}}(\alpha_p, \alpha_m) = |\alpha_p - \alpha_m|. \quad (4)$$

where $r_p, r_m \in \mathcal{S}$ are predicted and measured reflectances, $\mathbf{LAB} : \mathcal{S} \mapsto \text{CIELAB}$ is the function that computes CIELAB values from reflectances assuming specified viewing conditions, $\alpha_p, \alpha_m \in \mathcal{A}$ are predicted and measured translucency α -values [34]. The weights $a, b \in \mathbb{R}$ are hyper-parameters and are set as $a = 50$ and $b = 10$ according to the relative magnitudes and importance of the loss functions as described in [33].

As our core contribution, we show the methodology of how to inject prior knowledge of tonal-to-optical monotonicity relationships and smoothness into the PDL model by introducing extra losses. We do this with two examples of monotonicity constraints that a model should satisfy to avoid banding artifacts in separations.

3. Injecting prior knowledge of monotonic relationships

We propose injecting monotonicity constraints of tonal-optical relationships into the model by adding an extra loss that we refer to as *monotonicity loss*, to penalize positive derivatives. This was inspired by Liu *et al.* [43] who proposed a derivative-based approach to inject monotonicity to arbitrary neural networks. They considered monotonicity only in one dependent variable w.r.t. multiple independent variables, and applied *equivalent* loss-weights across all mapping relationships. Furthermore, they used a derivative normalization that causes the loss to vanish when a small number of violations remain in the model training. In contrast, our strategy considers multiple dependent variables w.r.t. multiple independent variables, automatically adjusts loss-weights for different mapping relationships, and uses a different derivative normalization to address the described issue of vanishing loss. We refer to the supplemental document for an experimental comparison between Liu *et al.* and our approach.

Note that monotonicity loss is calculated based on derivatives thus groundtruth data is not required. This allows calculating monotonicity loss on any training samples that are sampled from the whole input space. To enlarge data coverage, the samples can be re-sampled differently at each training iteration thus vary from iteration to iteration. This has an advantage that theoretically we have an infinite data pool for sampling and training, aiding model generalization.

In this paper, we propose injecting two monotonic relationships between tonal values and resulting visual quantities: 1. Increasing the fraction of black material in a material mixture does not increase lightness CIE- L^* , 2. Increasing the fraction of transparent material (a material with almost zero absorption and scattering) in a material mixture does not increase the translucency parameter α , where $\alpha = 0$ corresponds to a transparent material and $\alpha \approx 1$ to an opaque material [34]. Note that the proposed concept can be also used for other monotonic relationships between printing materials and optical/visual quantities identified a priori.

Specifically, to calculate the monotonicity loss for lightness CIE- L^* w.r.t. black material K , at each training iteration we select a random set of tonals $\mathcal{M} \subset \mathcal{T}$ and extract the subset \mathcal{M}_{LK} with

positive derivatives of CIE- L^* w.r.t. the black material, i.e.

$$\mathcal{M}_{LK} = \left\{ \tau \in \mathcal{M} \mid \left. \frac{\partial \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial t_K} \right|_{\mathbf{t}=\tau} > 0 \right\} \quad (5)$$

where $\mathbf{t} \in \mathcal{T}$, t_K is the element of \mathbf{t} corresponding to the black material, and $\mathbf{L} : \mathcal{S} \times \mathcal{A} \mapsto [0, 100]$ extracts lightness CIE- L^* from model predictions, i.e. it uses the predicted reflectance and computes lightness for a given viewing condition (illuminant, observer). In this paper, we use CIED50 as illuminant and the CIE1936 color matching functions as observer.

The monotonicity relationship between the black tonal t_K and lightness CIE- L^* is violated by the model for positive derivatives. Thus, only positive derivatives are considered in the loss function:

$$\mathbf{E}_{\text{mono_LK}}(\mathcal{M}_{LK}) = \frac{1}{|\mathcal{M}_{LK}|} \sum_{\tau \in \mathcal{M}_{LK}} \left. \frac{\partial \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial t_K} \right|_{\mathbf{t}=\tau} \quad (6)$$

where $|\mathcal{M}_{LK}|$ is the cardinality of \mathcal{M}_{LK} , i.e. the number of elements in the set.

Similarly, we induce the monotonic relationship between the transparent material and the translucency parameter α :

$$\mathcal{M}_{AT} = \left\{ \tau \in \mathcal{M} \mid \left. \frac{\partial \mathbf{A}(\mathbf{PDL}(\mathbf{t}))}{\partial t_T} \right|_{\mathbf{t}=\tau} > 0 \right\} \quad (7)$$

where $\mathbf{t} \in \mathcal{T}$, t_T is the element of \mathbf{t} corresponding to the transparent material, and $\mathbf{A} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{A}$ extracts translucency α -values from model predictions. The loss is then computed as follows

$$\mathbf{E}_{\text{mono_AT}}(\mathcal{M}_{AT}) = \frac{1}{|\mathcal{M}_{AT}|} \sum_{\tau \in \mathcal{M}_{AT}} \left. \frac{\partial \mathbf{A}(\mathbf{PDL}(\mathbf{t}))}{\partial t_T} \right|_{\mathbf{t}=\tau} \quad (8)$$

Modern deep learning tools allow conveniently computing derivatives of a neural network's output w.r.t. its input, e.g. via the *tf.GradientTape* API of TensorFlow [44].

4. Injecting smoothness heuristics

We assume that the printer's optical transfer function describing the forward relationship between tonal values and optical/visual quantities is smooth, i.e. it does not contain high-frequencies such as edges or bumps. Observed high-frequencies are rather measurement or printing noise and should not be considered by the optical printer model. Thus, our aim is to determine a model with maximally smooth predictions without significantly sacrificing accuracy.

For this, we propose a second-order derivative-based smoothing loss that we refer to as *Laplacian loss* because of its similarity with the Laplace operator. Even though this loss could be considered for all output dimensions of the optical printer model, we restrict it to operate on lightness CIE- L^* only, which is computed from the predicted reflectance for a given viewing condition. Considering just lightness minimizes computational effort and allows model optimization w.r.t the perceptually most relevant contrast-related quantity, since the human visual system's sensitivity to high-frequency achromatic contrasts is larger than to high-frequency chromatic contrasts [45,46].

The Laplacian loss is computed as:

$$\mathbf{E}_{\text{lap}}(\mathcal{M}) = \frac{1}{m|\mathcal{M}|} \sum_{\tau \in \mathcal{M}} \sum_{i=1}^m \log \left(\left| \left. \frac{\partial \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial t_i^2} \right|_{\mathbf{t}=\tau} \right| + 1 \right) \quad (9)$$

where m is the dimension of the tonal space and $\mathbf{t} = (t_1, \dots, t_m)^T \in \mathcal{T}$. Since the magnitude and not the sign of second-order derivatives is a measure of smoothness, we use their absolute value.

Very large second-order derivatives might impair learning by overshooting. Therefore, we take the logarithm to reduce such overshooting risk. We constrain the lower-bound of the logarithm to 0 by adding 1, so that vanishing second-order derivatives do not contribute to the loss.

Similarly as for the monotonicity losses, the samples to compute the Laplacian loss can be re-sampled differently at each training iteration.

5. Robust plausible deep learning (RPDL) optical printer model

5.1. Structure of the RPDL model

The RPDL model operates also on tonals and shares the basic structure with the PDL model [33] described in section 2.2. In contrast to the original PDL model [33], the number of neurons is smaller but sufficient to obtain similar results while reducing the network's capacity to overfit. Specifically, the trunk has 3 hidden layers each with 200, 300, and 300 neurons respectively. The reflectance-predicting branch has one hidden layer with 100 neurons and an output layer with 31 neurons to predict spectral reflectances $r \in \mathcal{S} = [0, 1]^{31}$. The translucency-predicting branch has one hidden layer with 30 neurons and an output layer with 1 neuron to predict translucency $\alpha \in \mathcal{A} = [0, 1]$. The RPDL model uses the same activation functions as the PDL model. The detailed network design is shown as a diagram in the supplemental document. Reducing the network's capacity was done as a hyper-parameter optimization on validation sets.

In our experiments this network capacity is also used for the PDL model because it does not adversely impact the model's prediction accuracy.

5.2. Loss function and hyper-parameter optimization

The loss function is a weighted average of PDL's loss \mathbf{E}_{PDL} and the extra losses defined by Eqs. (6), 8 and 9:

$$\mathbf{E} = \mathbf{E}_{\text{PDL}} + c_1 \mathbf{E}_{\text{mono_LK}} + c_2 \mathbf{E}_{\text{mono_AT}} + c_3 \mathbf{E}_{\text{lap}} \quad (10)$$

where weights $c_1, c_2, c_3 \in \mathbb{R}$ are hyper-parameters that are adjusted to the printer as follows: The weights are set to a very small value $\epsilon > 0$ (e.g. $\epsilon = 0.001$), so that the prediction accuracy on validation data is similar to the model that is just trained using \mathbf{E}_{PDL} [33]. Then, the weights are increased until the accuracy of the model on validation data starts decreasing. Specifically, we increase the weights by a factor of 3 until the average CIEDE2000 error between prediction and ground truth of validation data increases by 5% of the minimum error achieved so far. We refer to the supplemental document for more details on the optimization effort.

6. Experiments

6.1. Data sets

Our experiments use all the three datasets that were used in [33] to characterize state-of-the-art material-jetting 3D-printers employing six materials (Cyan (C), Magenta (M), Yellow (Y), Black(K), White(W), Clear(T)): One dataset to characterize a Stratasys J750 printer and two datasets to characterize two Mimaki 3DUJ-553 printers. The datasets consist of reflectance and α -measurements of printed flat targets with known tonal values, except for the "Mimaki 2" dataset where all samples are opaque and thus α -measurements were not collected and not available for experiments. We refer to [33] for details on the set of sampled tonal values and the measurement setup. We also collected a new dataset from a second Stratasys J750 printer using the same measurement procedure. We denote the first Stratasys dataset as *Stratasys 1* and the newly collected dataset as *Stratasys 2*. These two Stratasys datasets are obtained from two different J750 printers using the same materials. In addition to inter-machine variability the datasets deviate also in the function H used to transform tonals to material ratios (see Sec. 2.1).

Stratasys 2: We list the tonal values encoded in 8 bit. The dataset consists of a regular grid $\{0, 85, 170, 255\}^5 \subset \text{CMYKT}$ of tonal values, 976 random CMYKT-samples, 1500 random opaque CMYK-samples, i.e. $T = 0$, and 500 random CMYK-samples with $T = 255$. In total there are $4^5 + 976 + 1500 + 500 = 4000$ samples.

6.2. Computing and evaluating predictions

Similar to [33], 300 samples are held-out as the test set, and the remaining samples are split into a validation and a training set: The validation set consists of 10% of these samples to fit the hyper- and regularization parameters and the training set consists of the remaining 90% samples to fit the neural network weights. We refer to the union of these training and validation sets as *big data*, to distinguish from a much smaller data i.e. the *small data* described next. The *small data* consists of only 10% of the big data, and is used to investigate the influence of the proposed approach on a much smaller dataset. The training set always contains the Neugebauer primaries. Small and big data for *Mimaki 1*, *Mimaki 2* and *Stratasys 1* were selected similarly as in [33]. In addition to the Neugebauer primaries, small data for *Stratasys 2* contains randomly-selected samples from the aforementioned 976 random CMYKT-samples described in section 6.1. To compute predictions for PDL or RPD models, we averaged 10 predictions computed by respective models trained on different decompositions in validation and training sets. We report in this paper these so-called *10-fold* predictions, unless explicitly specify *1-fold*. Note that for computing *10-fold* predictions all 10 models are trained on exactly the same training data.

6.3. Software and hardware setup

Our model is implemented with TensorFlow 2.2.0, and is trained on an NVIDIA GeForce RTX 2080 SUPER GPU and an NVIDIA GeForce RTX 3090 GPU. Training a RPD model takes approx. 420s on 2825 training samples and 170s on 282 samples. Predicting on 300 samples takes approx. 0.005s.

6.4. Training method

The neural network structure described in section 2.2 is used for both the PDL and RPD models, but the proposed extra losses (i.e. monotonicity loss and Laplacian loss) are used only for the RPD models. The initial learning rate is 0.003, with the same learning rate decay used in [33]. The original losses [33] are calculated on the fixed training samples for both models, while the extra losses are calculated on extra training samples (without using the measurements). These extra data is re-sampled randomly from the whole input space at each training iteration, thus is varying from iteration to iteration. Specifically, at each iteration, 2000 samples are uniformly randomly sampled from tonal space to compute the monotonicity losses, and similarly another 2000 samples for the Laplacian loss. *Early stopping* [47] and *dropout* [48] regularization strategies are employed to avoid overfitting as described in [33] for both approaches.

6.5. Similar accuracies of the PDL and RPD models

Table 1 summarizes the accuracy comparison between the PDL approach [33] and the proposed RPD approach. We mark in bold face those model results that outperform the counterparts by $> 5\%$ w.r.t. color accuracy. The results show that the proposed RPD approach leads to similar accuracy on big data. For small data, RPD results in an accuracy improvement of at least 5% in all cases (about 8% smaller average CIEDE2000 errors for *Stratasys 1*, *Stratasys 2*, and *Mimaki 2*) except for the *Mimaki 1* average color error, for which we see still an improvement but less than 5%. Even though our objective is not to improve accuracy, but to achieve better robustness and plausibility preserving accuracy, the improved accuracy on small data indicates that the RPD model has a better generalization ability than the PDL model. The 90th percentile

Table 1. Model accuracy comparison between PDL and RPDL (A format like "0.44/0.79" means average error is 0.44 and the 90-th percentile error is 0.79.)

	Stratasys 1		Stratasys 2		Mimaki 1		Mimaki 2	
	big data	small data	big data	small data	big data	small data	big data	small data
PDL	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}
	0.437/0.785	2.26/4.02	1.16/2.00	2.04/3.71	0.794/1.40	2.85/6.07	ΔE_{00}	ΔE_{00}
	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	1.17/2.37	1.90/3.73
	0.0042/0.0090	0.0123/0.0291	0.0090/0.0183	0.0114/0.0234	0.0061/0.0118	0.0202/0.0486		
RPDL	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}	ΔE_{00}
	0.452/0.805	2.07/3.83	1.10/1.90	1.89/3.30	0.810/1.38	2.81/5.73	ΔE_{00}	ΔE_{00}
	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	$\Delta\alpha$	1.18/2.38	1.75/3.51
	0.0046/0.0104	0.0106/0.0236	0.0087/0.0175	0.0113/0.0261	0.0069/0.0142	0.0209/0.0480		

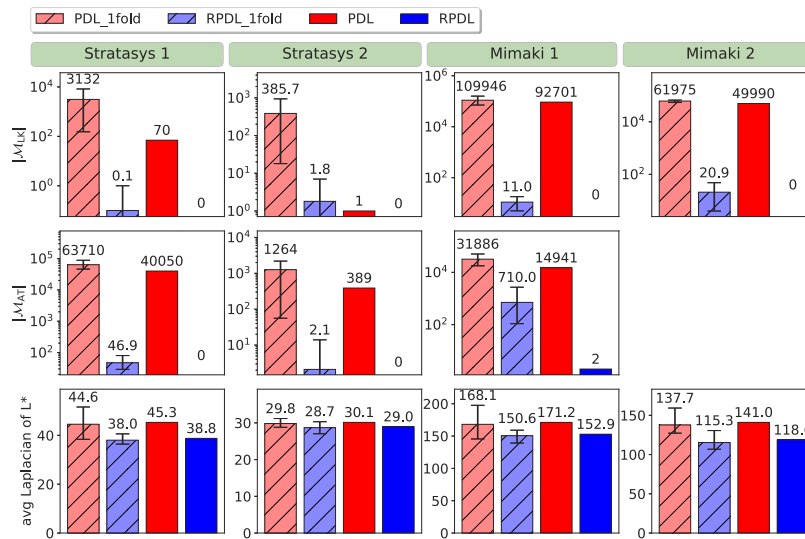


Fig. 6. Bar plots of monotonicity violations (L^* -vs-K and α -vs-T) and average Laplacian (original definition). Each quantitative result is shown above the corresponding bar. For 1-fold models, each shown quantitative result is the averaged performance through 10 models, with a whisker to show the maximum and the minimum (Note for some bars for the 1-fold RPDL, the minimum is zero thus can't be displayed due to log scale in y-axis). Numerical results larger than 1000 are rounded to integer.

α -errors of both strategies are below the just noticeable difference (approx. $\Delta\alpha = 0.1$ [34]) even on small data, which means the α -accuracy is already sufficient for most applications.

6.6. Improvements in robustness and plausibility

We further evaluate the PDL and RPDL models with respect to violations in L^* -vs-K and α -vs-T monotonicity, as well as smoothness. For this, we randomly selected one million samples from the tonal space $\mathcal{M} \in \mathcal{T}$ and compute $\mathcal{M}_{LK} \subset \mathcal{M}$ and $\mathcal{M}_{AT} \subset \mathcal{M}$ according to Eq. (5) and 7, respectively. The number of violations are reported by the positive derivatives, i.e. the cardinality of \mathcal{M}_{LK} and \mathcal{M}_{AT} , respectively. Smoothness is evaluated by computing the average magnitude of the Laplacian for L^* using the original definition, i.e. without the log-attenuation as in the loss function (Eq. (9)). Figure 6 shows the quantitative results, and we make the following observations:

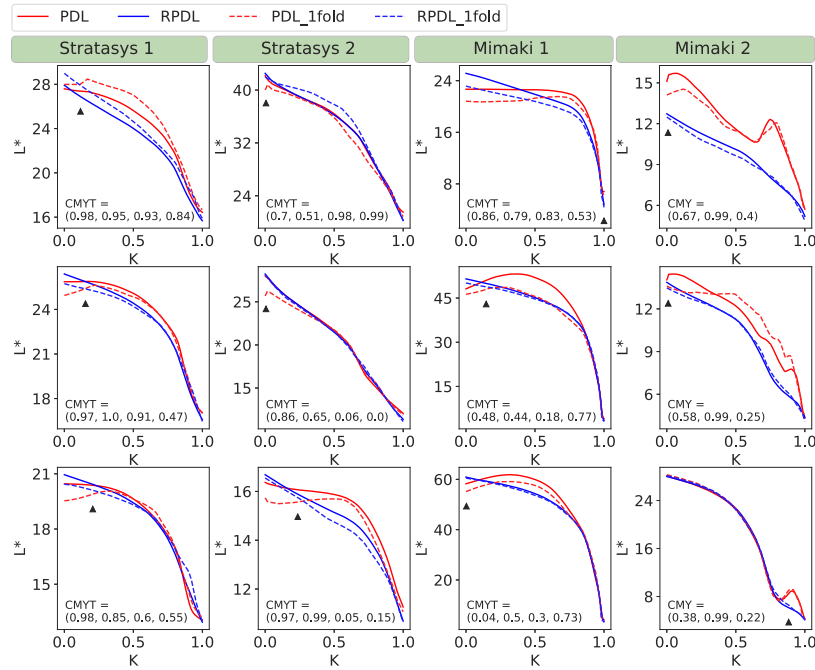


Fig. 7. RPDL model resolves L^* -vs- K violations in PDL model. A small black arrow is used to indicate the violation's location.

1. The models evaluated on the Stratasys datasets show many more monotonicity violations in α -vs- T than in L^* -vs- K , while the models computed on the Mimaki datasets have a much larger number of violations for both relationships.
2. Model averaging tends to reduce monotonicity violations but does not always work well: e.g. for Mimaki 1, the PDL 10-fold model has a similarly large number of L^* -vs- K violations compared to the PDL 1-fold model. On the other hand, the RPDL 1-fold model already drastically reduces the number of L^* -vs- K violations to 11 which is a much bigger reduction compared to that by model averaging. Furthermore, the 10-fold version of RPDL completely removes violations in both L^* -vs- K and α -vs- T for all the 4 datasets, except for in α -vs- T for Mimaki 1 where there are 2 violations. However, the 2 violations have positive derivatives below 10^{-5} which is significantly smaller compared to up to 25 from the PDL 10-fold model.
3. RPDL always reduces the average Laplacian, e.g. by up to 16% compared to PDL for Mimaki 2, indicating much smoother predictions than PDL.

Figure 7 shows three typical L^* -vs- K monotonicity violations for each of the four datasets with the other tonals held constant. It shows the PDL has L^* -vs- K violations and bumpiness, and model averaging relieves the issues but fails to completely resolve them, while the RPDL fully removes them all. Figure 8 shows a similar trend in α -vs- T (Note the Mimaki 2 dataset is not available for the α -vs- T analysis since it only includes fully opaque prints). Note that the RPDL achieves better robustness and plausibility without sacrificing accuracy as summarized in Table 1.

Figure 1 illustrates color ramps with L^* -vs- K violations predicted by PDL on the Mimaki 2 dataset. The figure shows that the PDL model overfits to the erroneous data violating monotonicity, while despite the data outlier the RPDL model ensures monotonicity indicating better robustness.

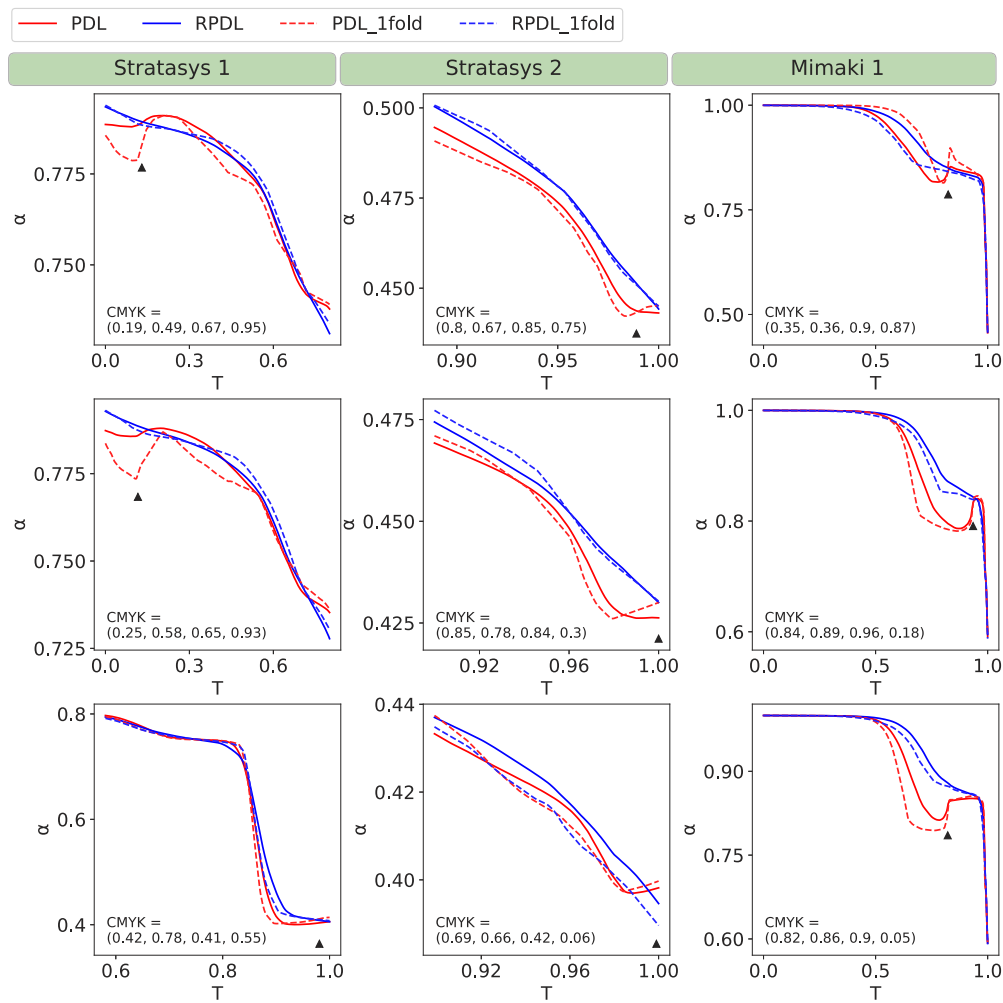


Fig. 8. RPDL model resolves α -vs- T violations in PDL model

Figure 9 shows a color ramp with L^* -vs- K monotonicity violations predicted by the 1-fold PDL model on the Stratasys 1 dataset. It shows that the PDL-predicted ramp slightly increases in L^* for increasing K values in the range [0.9, 1.0], i.e. 10% of the K scale, and has a sharp turning point at $K = 0.9$, while the 1-fold RPDL ensures monotonicity with improved smoothness.

Figure 2 visualizes predicted L^* values as a function of K and Y tonals with other materials constant. It shows that the PDL model has bumpy 3D surface and skewed contours violating L^* -vs- K monotonicity, while the RPDL model has much smoother predictions ensuring monotonicity.

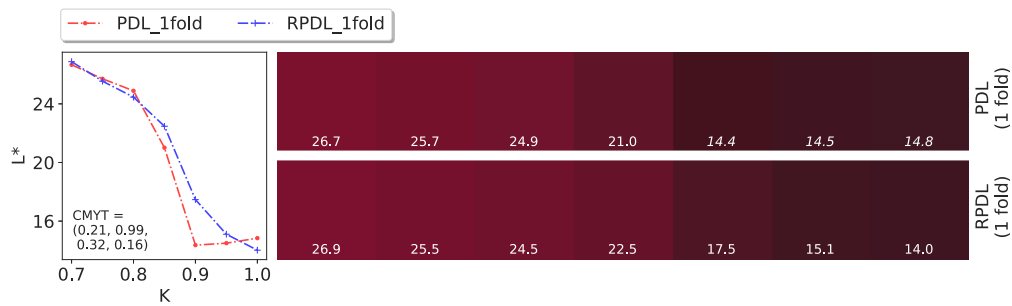


Fig. 9. L^* as a function of K with other materials fixed on a tonal case of the *Stratasys 1* dataset, and the resulting colors corresponding to the K values.

7. Conclusion and future work

To address physically-implausible and noisy predictions from the Pure-Deep-Learning (PDL) optical printer model [33], we propose a methodology to induce physical heuristics into the model via new loss functions that do not rely on additional printed samples for training: a derivative-based monotonicity loss to induce *a priori* knowledge of tonal-to-optical monotonicity relationships, as well as a Laplacian-based smoothness loss to induce smoothness. For the monotonicity relationships we select L^* -vs.- K and α -vs.- T . We introduce the Robust Plausible Deep Learning (RPDL) optical model via a learning strategy by combining these loss functions with PDL's original loss functions using an automatic hyper-parameter optimization considering an upper threshold for color accuracy losses. Our experiments on four state-of-the-art 6-material 3D printers show that the RPD optical model is more robust to data outliers and creates much smoother predictions ensuring monotonicity. The improvement in robustness and plausibility does not sacrifice accuracy, and yields even up to 8% higher accuracy on small training data indicating a better generalization ability.

This approach can be extended canonically to induce other tonal-optical monotonicity relationships into the model. Future work shall focus on exploring new monotonicity relationships and investigating how inducing such prior knowledge into the model can improve its generalization ability to further reduce the number of training samples without worsening accuracy. For instance, spectral monotonicity relationships can be explored: A printing material having the smallest/biggest reflectance factor of all available printing materials for a distinct wavelength must decrease/increase the reflectance factor for this wavelength if its fraction increases in the material mixture (assuming non-fluorescent materials). This applies to colored printing materials if the fractions of white and black materials in the material mixture are kept constant.

Funding. H2020 Marie Skłodowska-Curie Actions (813170); Allianz Industrie Forschung (20852 N/2).

Acknowledgments. We thank Alan Brunton and Johann Reinhard for their advice and support.

Disclosures. The authors declare no conflicts of interest.

Data availability. Datasets of Stratasys 1, Mimaki 1, and Mimaki 2 are available via the PDL paper [33]. The new dataset, i.e. Stratasys 2, is not publicly available at this time but may be obtained from the authors upon reasonable request.

Supplemental document. See [Supplement 1](#) for supporting content.

References

1. A. Brunton, C. A. Arikan, and P. Urban, "Pushing the limits of 3d color printing: Error diffusion with translucent materials," *ACM Trans. Graph.* **35**(1), 1–13 (2015).
2. C. A. Arikan, A. Brunton, T. M. Tanksale, and P. Urban, "Color-managed 3d-printing with highly translucent printing materials," in *SPIE/IS&T Electronic Imaging Conference*, (San Francisco, 2015), pp. 9398 – 9398–9.

3. D. Sumin, T. Rittig, V. Babaei, T. Nindel, A. Wilkie, P. Didyk, B. Bickel, J. Křivánek, K. Myszkowski, and T. Weyrich, "Geometry-aware scattering compensation for 3d printing," *ACM Trans. Graph.* **38**(4), 1–14 (2019).
4. M. Hašan, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz, "Physical reproduction of materials with specified subsurface scattering," *ACM Trans. Graph.* **29**(4), 61 (2010).
5. Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo, "Fabricating spatially-varying subsurface scattering," *ACM Trans. Graph.* **29**(4), 1–10 (2010).
6. A. Brunton, C. A. Arkan, T. M. Tanksale, and P. Urban, "3d printing spatially varying color and translucency," *ACM Trans. Graph.* **37**(4), 1–13 (2018).
7. A. Murray, "Monochrome reproduction in photoengraving," *J. Franklin Inst.* **221**(6), 721–744 (1936).
8. H. E. J. Neugebauer, "The Theoretical Basis of Multicolor Letterpress Printing (Translated D. Wyble and A. Kraushaar)," *Color Res. Appl.* **30**(5), 322–331 (2005).
9. R. Rolleston and R. Balasubramanian, "Accuracy of Various Types of Neugebauer Model," in *IS&T/SID*, (Scottsdale Ariz., 1993), pp. 32–36.
10. J. A. C. Yule and W. J. Nielsen, "The penetration of light into paper and its effect on halftone reproduction," in *TAGA Proceedings*, vol. 4 (1951), pp. 65–76.
11. J. A. C. Yule and R. S. Colt, "Colorimetric Investigations in Multicolor Printing," in *TAGA Proceedings*, (1951), pp. 77–82.
12. J. Viggiano, "Modeling the Color of Multi-color Halftones," in *TAGA Proceedings*, (1990), pp. 44–62.
13. R. Hersch, P. Emmel, F. Collaud, and F. Crété, "Spectral reflection and dot surface prediction models for color halftone prints," *J. Electron. Imaging* **14**(3), 033001 (2005).
14. R. Hersch and F. Crété, "Improving the yule-nielsen modified spectral neugebauer model by dot surface coverages depending on the ink superposition conditions," in *Proc. SPIE*, vol. 5667 (2005), pp. 434–445.
15. B. D. Hensley and J. A. Ferwerda, "Colorimetric characterization of a 3d printer with a spectral model," in *Color and Imaging Conference*, (Society for Imaging Science and Technology, 2013), pp. 160–166.
16. F. Clapper and J. Yule, "Reproduction of color with halftone images," in *TAGA Proceedings*, (1955), pp. 1–14.
17. F. Clapper and J. Yule, "The effect of multiple internal reflections on the densities of half-tone prints on paper," *J. Opt. Soc. Am.* **43**(7), 600–603 (1953).
18. R. Hersch, F. Collaud, and P. Emmel, "Reproducing color images with embedded metallic patterns," *ACM Trans. Graph.* **22**(3), 427–434 (2003).
19. G. Rogers, "A generalized clapper–yule model of halftone reflectance," *Color Res. Appl.* **25**(6), 402–407 (2000).
20. M. Hébert and R. D. Hersch, "Review of spectral reflectance models for halftone prints: principles, calibration, and prediction accuracy," *Color Res. Appl.* **40**(4), 383–397 (2015).
21. A. U. Agar and J. P. Allebach, "An iterative cellular ysn method for color printer characterization," in *IS&T/SID*, (Scottsdale Ariz., 1998), pp. 197–200.
22. V. Babaei and R. D. Hersch, "*n*-ink printer characterization with barycentric subdivision," *IEEE Trans. on Image Process.* **25**(7), 3023–3031 (2016).
23. P. Kubelka and F. Munk, "Ein Beitrag zur Optik der Farbanstriche," *Zeitschrift für Technische Physik* **12**, 593–601 (1931).
24. J. Saunderson, "Calculation of the color of pigmented plastics," *J. Opt. Soc. Am.* **32**(12), 727–729 (1942).
25. T. P. Van Song, C. Andraud, and M. V. Ortiz Segovia, "Implementation of the four-flux model for spectral and color prediction of 2.5 d prints," in *NIP & Digital Fabrication Conference*, vol. 2016 (IS&T, 2016), pp. 26–30.
26. T. P. Van Song, C. Andraud, and M. V. Ortiz-Segovia, "Towards spectral prediction of 2.5 d prints for soft-proofing applications," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, (IEEE, 2016), pp. 1–6.
27. L. Simonot, R. D. Hersch, M. Hébert, and S. Mazaauric, "Multilayer four-flux matrix model accounting for directional-diffuse light transfers," *Appl. Opt.* **55**(1), 27–37 (2016).
28. C. J. Zoller, A. Hohmann, F. Forschum, S. Geiger, M. Geiger, T. P. Ertl, and A. Kienle, "Parallelized monte carlo software to efficiently simulate the light propagation in arbitrarily shaped objects and aligned scattering media," *J. Biomed. Opt.* **23**(06), 1 (2018).
29. S. Tominaga, "Color control using neural networks and its application," in *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, vol. 2658 (International Society for Optics and Photonics, 1996), pp. 253–260.
30. S. Abet and G. Marcu, "A neural network approach for rgb to ymck color conversion," in *TENCON'94-1994 IEEE Region 10's 9th Annual International Conference on: 'Frontiers of Computer Technology'*, (IEEE, 1994), pp. 6–9.
31. D. Littlewood, P. Drakopoulos, and G. Subbarayan, "Pareto-optimal formulations for cost versus colorimetric accuracy trade-offs in printer color management," *ACM Trans. Graph.* **21**(2), 132–175 (2002).
32. L. Shi, V. Babaei, C. Kim, M. Foshey, Y. Hu, P. Sitthi-Amorn, S. Rusinkiewicz, and W. Matusik, "Deep multispectral painting reproduction via multi-layer, custom-ink printing," *ACM Trans. Graph.* **37**(6), 1–15 (2018).
33. D. Chen and P. Urban, "Deep learning models for optically characterizing 3d printers," *Opt. Express* **29**(2), 615–631 (2021).
34. P. Urban, T. M. Tanksale, A. Brunton, B. M. Vu, and S. Nakauchi, "Redefining A in RGBA: Towards a standard for graphical 3d printing," *ACM Trans. Graph.* **38**(3), 1–14 (2019).
35. S. Tsutsumi, M. Rosen, and R. Berns, "Spectral Reproduction Using LabPQR: Inverting the Fractional-Area-Coverage-to-Spectra Relationship," in *ICIS*, (IS&T, Rochester, NY, 2006), pp. 107–110.

36. P. Urban and R.-R. Grigat, "Spectral-Based Color Separation using Linear Regression Iteration," *Color Res. Appl.* **31**(3), 229–238 (2006).
37. P. Urban, M. R. Rosen, and R. S. Berns, "Accelerating Spectral-Based Color Separation within the Neugebauer Subspace," *J. Electron. Imaging* **16**(4), 043014 (2007).
38. P. Urban and M. R. Rosen, "Inverting the Cellular Yule-Nielsen modified Spectral Neugebauer Model," in *Ninth International Symposium on Multispectral Color Science and Application*, (Taipei, Taiwan, 2007), pp. 29–35.
39. J. Morovič, A. Albarran, J. Arnabat, Y. Richard, and M. Maria, "Accuracy-preserving smoothing of color transformation luts," in *Color and Imaging Conference*, vol. 2008 (Society for Imaging Science and Technology, 2008), pp. 243–246.
40. J. Zhang, H. Nachlieli, D. Shaked, S. Shiffman, and J. P. Allebach, "Psychophysical evaluation of banding visibility in the presence of print content," in *Image Quality and System Performance IX*, vol. 8293 (International Society for Optics and Photonics, 2012), p. 82930S.
41. D. R. Wyble and R. S. Berns, "A Critical Review of Spectral Models Applied to Binary Color Printing," *Color Res. Appl.* **25**(1), 4–19 (2000).
42. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30 (2013), p. 3.
43. X. Liu, X. Han, N. Zhang, and Q. Liu, "Certified monotonic neural networks," in *Advances in Neural Information Processing Systems*, vol. 33 H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds. (Curran Associates, Inc., 2020), pp. 15427–15438.
44. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015), [tensorflow.org](https://www.tensorflow.org).
45. F. L. Van Nes and M. A. Bouman, "Spatial modulation transfer in the human eye," *J. Opt. Soc. Am.* **57**(3), 401–406 (1967).
46. K. T. Mullen, "The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings," *The J. Physiol.* **359**(1), 381–400 (1985).
47. I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1 (MIT Press Cambridge, 2016).
48. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(56), 1929–1958 (2014).