Research Article Vol. 31, No. 8 / 10 Apr 2023 / Optics Express 13486

Optics EXPRESS

Multi-printer learning framework for efficient optical printer characterization

DANWU CHEN[1,2,*] AND PHILIPP URBAN[1,3]

[1] Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany
[2] Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany
[3] Norwegian University of Science and Technology NTNU, Gjøvik, Norway
[*] danwu.chen@igd.fraunhofer.de

Abstract: A high prediction accuracy of optical printer models is a prerequisite for accurately reproducing visual attributes (color, gloss, translucency) in multimaterial 3D printing. Recently, deep-learning-based models have been proposed, requiring only a moderate number of printed and measured training samples to reach a very high prediction accuracy. In this paper, we present a multi-printer deep learning (MPDL) framework that further improves data efficiency utilizing supporting data from other printers. Experiments on eight multi-material 3D printers demonstrate that the proposed framework can significantly reduce the number of training samples thus the overall printing and measurement efforts. This makes it economically feasible to frequently characterize 3D printers to achieve a high optical reproduction accuracy consistent across different printers and over time, which is crucial for color- and translucency-critical applications.

1. Introduction

Reproducing visual attributes (color, gloss, or translucency) via 3D printing requires optical printer models that accurately predict optical properties of the printed output from the input signals controlling the printer. Such signals can be material ratios or arrangements as well as print-process parameter selections.

Optical printer models can be classified into phenomenological models [1–16], models based on the radiative transfer equation (RTE) or its simplifications [17–22], and neural-network-based models [23–28]. A discussion of these models can be found in our paper [27]. The process of fitting the parameters of an optical model using printed and measured samples is called *optical characterization*. The objective of developing a printer model is to maximize prediction accuracy while minimizing optical characterization effort.

In this paper, we propose a framework that uses existing optical characterization data from multiple printers to improve the prediction accuracy of deep-learning-based models on a *targeted* printer. This approach improves accuracy without requiring more data from the targeted printer. In turn, it needs drastically less data to achieve a similar accuracy. To our knowledge, this is the first approach for optically modeling a printer using characterization data from other printers, even though these printers might differ from the targeted printer significantly as described in the following.

We denote by a *printing system* the combination of hardware, software and used printing materials. Even if controlled with equal input signals, different 3D printing systems produce outputs that can significantly differ in appearance. This is not surprising because

1. printing material sets can significantly differ in intrinsic optical (refractive index, scattering, absorption, phase function) and mechanical properties (viscosity, surface tension), impacting light transport and physical material mixing behavior,

2. the software can use halftoning algorithms resulting in different material arrangements, or interlacing strategies for material placement,

#487526 https://doi.org/10.1364/OE.487526
Journal © 2023 Received 10 Feb 2023; revised 16 Mar 2023; accepted 17 Mar 2023; published 7 Apr 2023

3. the hardware may differ in material placement accuracy or other factors such as UV exposure in material jetting.

For these reasons, an optical printer model fitted to one printing system performs usually poorly if applied to a different printing system.

On the other hand, even two instances of the *same* printing system (e.g. two printers using the same material set, software and hardware, but located at two different geographic locations) may create deviating outputs for the same control signal because of material lot variability, different maintenance state or deviating environmental factors (humidity or temperature). We call it *inter-printer variability*. These factors may also cause a printing system to deviate over time, which we call *intra-printer variability*. Optical printer models must be re-fitted to account for inter- and intra-printer variability to ensure maximum prediction performance.

Yet another important factor causing deviations in the visual appearance of the printed object is *post-processing*, such as polishing, clear coating or applying chemical treatments. Fig. 1 shows the color difference histogram of the same set of physical samples measured before and after applying a silicone-based plastic care treatment that seals the surface of the prints preventing them from drying out. The average color difference between post-processed and non-processed prints is large, particularly for dark samples. Existing optical printer models do not model postprocess treatments. They consider them only implicitly because their parameters are fitted to printed and measured samples on which the postprocess has been applied. If the postprocess changes, the models must be re-fitted, requiring a complete new set of printed, post-processed and measured samples.

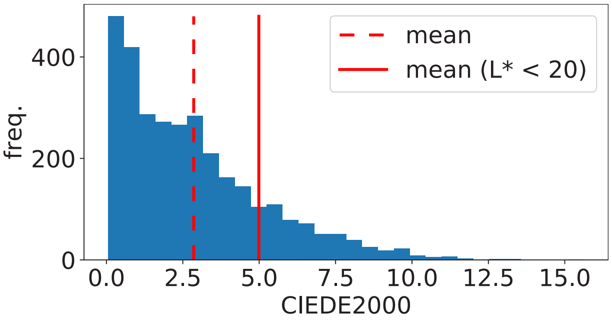

Fig. 1. Histogram of CIEDE2000 color differences between printed samples with and without applying plastic care post-process treatment. The dashed line represents the average color difference, and the solid line represents the average color difference for dark samples with $L^*<20$.

For the sake of simplicity, in the following we will use the term *different printers* to mean all these settings: *different printing systems*, *different instances of the same printing system* and *printing systems with deviating post-process treatments*.

Despite all the factors of different printers causing the appearance of their outputs to deviate, various optical and mechanical mechanisms are shared across such prints. These similarities may include physics of light transport or the volumetric material arrangement for prints which only deviate in postprocess treatments. The similarities dominate for similar printing systems even though the inter- and intra-printer variability might be noticeable. This leads to the open research question of how to exploit data from other printing systems possessing these similarities to 1) improve an optical model's prediction accuracy for a particular printer, or 2) to reduce the characterization effort by using fewer training samples without sacrificing accuracy.

In this paper, we propose a methodology answering this research question. We make the following contributions:

1. A dataset embedding-based learning framework to improve prediction accuracy and data efficiency of deep-learning optical printer models, by using existing characterization data from different printers.

2. A contrastive learning-based approach to learn dataset embeddings that are used as extra inputs (besides material arrangement or tonals) of the above dataset embedding-based learning framework.

3. A learning strategy for training the model and fitting its hyperparameters.

The proposed methodology can drastically reduce the number of required samples that have to be printed, measured and postprocessed to characterize a targeted printing system for achieving an application-specific prediction accuracy. This makes it economically feasible to continuously (e.g. every month/season instead of every one/several years) characterize machineries of 3D printers to achieve more consistent output across different printers and over time with smaller tolerances, which is crucial for color- and translucency-critical applications such as 3D printed prosthetic eyes or dental restorations.

2. Precondition and problem formulation

In this paper, an optical printer model is a function $\Phi : \mathcal{T} \mapsto \mathcal{S} \times \mathcal{A}$ that predicts spectral reflectances $\mathbf{r} \in \mathcal{S} = [0,1]^N$ and translucency [29] $\alpha \in \mathcal{A} = [0,1]$ from material arrangement i.e. tonals $\mathbf{t} \in \mathcal{T} = [0,1]^M$, $|\mathbf{t}|_1 \leq 1$. N is the number of considered wavelengths that is set to $N = 31$ with equidistant sampling in the range of [400nm, 700nm]. A 3D printing system using $M + 1$ printing materials can be controlled by an M-dimensional tonal vector, because the fraction of the $(M + 1)$-th material in the material mixture is implicitly defined by $1 - |\mathbf{t}|_1$. Here we use $M = 5$ corresponding to the materials cyan (C), magenta (M), yellow (Y), black (K), and clear (T). The implicitly defined material is white (W), i.e. for a printer using CMYKWT materials, $\mathbf{t} = (0,0,0,0,0)$ corresponds to the pure white ink. This approach can be generalized to more and other materials. As described in our previous works [27,28], different printing systems have different transforms to convert nominal tonals to effective tonals (e.g. 1D-per-tonal curves as described in [30]) for final printing. The previous works simply used the nominal tonals and this was not an issue because the optical printer characterizations for different printers are independent. However, in the setting of cross-printer generalization, using the same effective tonal space will reduce the input discrepancy among different printers. For this reason, we use effective tonals.

To characterize a new printer, existing datasets from other printing systems could be helpful in addition to the dataset collected for the targeted printer. Suppose there are m datasets $\{\mathcal{D}_i\}_{i=1}^m$ collected from m printers respectively. $\mathcal{D}_i$ is the dataset from the ith printer, consisting of n_i samples, i.e. $\mathcal{D}_i = \{(\mathbf{t}_i^{(j)}, \mathbf{r}_i^{(j)}, \alpha_i^{(j)})\}_{j=1}^{n_i}$, where $\mathbf{t}_i^{(j)} \in \mathcal{T} = [0,1]^M$ is the tonal vector of the jth sample in $\mathcal{D}_i$, $\mathbf{r}_i^{(j)} \in \mathcal{S} = [0,1]^N$ is the corresponding spectral reflectances vector, and $\alpha_i^{(j)} \in \mathcal{A} = [0,1]$ is the corresponding translucency α-value. Without loss of generality, we regard the Γth $(1 \leq \Gamma \leq m)$ printer as the targeted printer that we want to characterize using its dataset $\mathcal{D}_\Gamma$, as well as all datasets from other printers i.e. $\{\mathcal{D}_i | i \neq \Gamma\}_{i=1}^m$ as supporting data. In practice, the supporting datasets are from historical characterizations of other printers.

To facilitate reading, we provide a symbol lookup table in Appendix A.

3. Dataset embedding-based multi-printer learning framework

We propose a dataset embedding-based Multi-Printer Deep Learning Framework (or MPDL) to improve characterization accuracy and data efficiency, utilizing data from other printers. In the MPDL framework, each dataset $\mathcal{D}_i$ is implicitly represented by a pre-learnt vector (referred to as *dataset embedding*) $\mathbf{e}_i \in \mathcal{E} = \mathbb{R}^K$ (e.g. K=256). The dataset embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$ ($i \neq j$) for 2 dissimilar printers $\mathcal{D}_i$ and $\mathcal{D}_j$ should be different enough in order to help discriminate printers, while for similar printers the embeddings should be similar to reflect printer similarity. Dataset embedding $\mathbf{e}_i \in \mathcal{E}$ are fed into a single neural network to discriminatively learn diverse optical and mechanical mechanisms of multiple printers simultaneously. For this, the optical printer model's input domain is extended by the dataset embeddings $\mathcal{E}$, i.e. $\Omega_\theta : \mathcal{T} \times \mathcal{E} \mapsto \mathcal{S} \times \mathcal{A}$, where Ω_θ is an extended neural network-based optical printer model and θ is the neural network's parameters to be trained.

The proposed MPDL framework is general and not limited to a certain way of obtaining dataset embedding $\mathbf{e}_i \in \mathcal{E}$. Nevertheless, in Section 4 we will propose a contrastive learning-based approach to learn such dataset embeddings.

3.1. Dataset-aware feature learning

To allow the neural network to learn not only printer-dependent mechanisms but also printer-independent behavior (considering underlying similarities among printers), as shown in Fig. 2, we designed a 2-path feature learning: A path with hidden layers to learn dataset-aware features from dataset embedding $\mathbf{e}_i$ shown in Fig. 2(I), and another path with hidden layers to learn dataset-agnostic features from tonals $\mathbf{t}$ shown in Fig. 2(II). The learnt features from the 2 paths are merged via concatenation and fed to a sub-network shown in Fig. 2(III) that predicts the final appearance (translucency, spectral, and CIELAB color). The dataset-agnostic path and the dataset-aware path both consist of one hidden layer with 200 neurons, so the 2-path merged features contains 400 entries. The sub-network has the same structure as PDL [27] or RPDL [28] models, except that the first hidden layer of PDL and RPDL is discarded and that the sub-network takes as input the aforementioned 2-path merged features instead of tonals directly. Specifically, the trunk of the sub-network has 2 hidden layers each with 300 neurons. The reflectance-predicting branch has one hidden layer with 100 neurons and an output layer with 31 neurons to predict spectral reflectances $\mathbf{r} \in \mathcal{S} = [0, 1]^{31}$. The translucency-predicting branch has one hidden layer with 30 neurons and an output layer with 1 neuron to predict translucency $\alpha \in \mathcal{A} = [0, 1]$. Note that PDL and RPDL take as input the dataset-agnostic tonals only and thus are not able to perform dataset-aware feature learning, i.e. are not able to discriminatively learn diverse printer behaviors.

The dataset-agnostic path, the dataset-aware path, and the sub-network are trained together in an end-to-end manner with tonals and dataset embeddings as inputs, and with appearance (translucency, spectral, and CIELAB color) as output.

The whole framework contains around 0.3 million neural network parameters. We show a more detailed structure of the neural network in Section 1 of Supplement 1.

3.2. Dataset embedding-based adaptive loss weights

The state-of-the-art deep learning optical model, RPDL [28], used a loss that is a weighted summation of 6 loss terms:

$$\mathbf{E}_{\text{RPDL}} = \mathbf{E}_{\text{col}} + a_1\mathbf{E}_{\text{ref}} + a_2\mathbf{E}_{\text{tra}} + a_3\mathbf{E}_{\text{mono\_LK}} + a_4\mathbf{E}_{\text{mono\_AT}} + a_5\mathbf{E}_{\text{lap}} \tag{1}$$

where $\mathbf{E}_{\text{col}}$ is the CIEDE2000 color error computed for specific viewing conditions (illuminant, observer), $\mathbf{E}_{\text{ref}}$ is the spectral reflectances root-mean-square error (RMSE), $\mathbf{E}_{\text{tra}}$ is the α-based translucency absolute error, $\mathbf{E}_{\text{mono\_LK}}$ is monotonicity loss for lightness CIE-L^* w.r.t. black material K, $\mathbf{E}_{\text{mono\_AT}}$ is monotonicity loss for translucency α w.r.t. transparent material T, and $\mathbf{E}_{\text{lap}}$ is Laplacian smoothness loss. RPDL adopted the first three loss terms from PDL [27]. Weights $a_{1-5} \in \mathbb{R}$ are printer-agnostic hyper-parameters and are selected as described in [28], in particular $a_1 = 50, a_2 = 10, a_3 = 0.1, a_4 = 0.1$, and $a_5 = 10^{-5}$. For more details on loss terms and automatic hyper-parameter tuning we refer to [28]. We also list all loss terms in Appendix B.

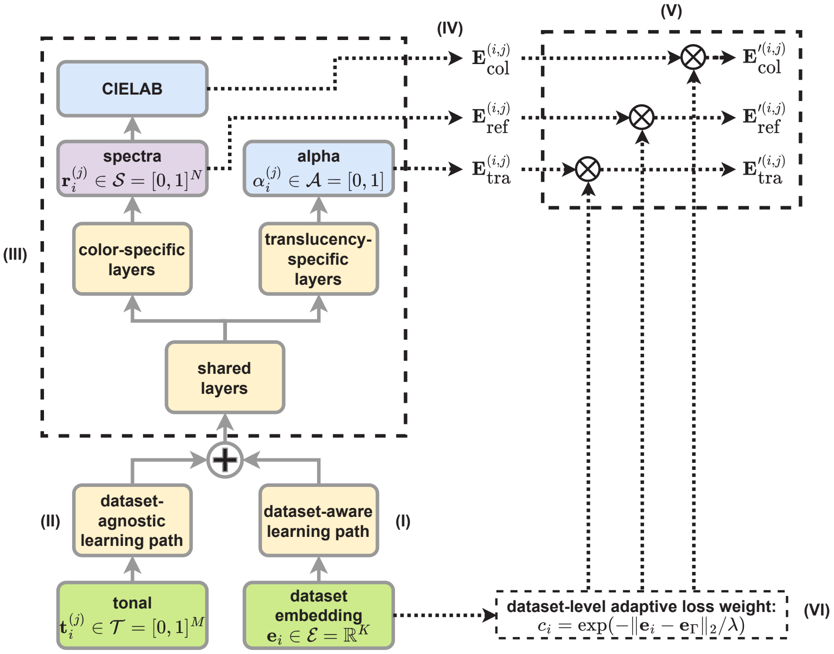
(IV)
(V)
(III)
CIELAB
spectra
$\mathbf{r}_i^{(j)} \in \mathcal{S} = [0,1]^N$
alpha
$\alpha_i^{(j)} \in \mathcal{A} = [0,1]$
color-specific layers
translucency-specific layers
shared layers
$\mathbf{E}_{\text{col}}^{(i,j)}$
$\mathbf{E}_{\text{ref}}^{(i,j)}$
$\mathbf{E}_{\text{tra}}^{(i,j)}$
$\mathbf{E}_{\text{col}}^{\prime(i,j)}$
$\mathbf{E}_{\text{ref}}^{\prime(i,j)}$
$\mathbf{E}_{\text{tra}}^{\prime(i,j)}$
(II)
dataset-agnostic learning path
dataset-aware learning path
(I)
tonal
$\mathbf{t}_i^{(j)} \in \mathcal{T} = [0,1]^M$
dataset embedding
$\mathbf{e}_i \in \mathcal{E} = \mathbb{R}^K$
dataset-level adaptive loss weight:
$c_i = \exp(-\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2/\lambda)$
(VI)

Fig. 2. Overview of the proposed dataset embedding-based learning framework. A dataset-aware learning path (I) that learns from dataset embedding $\mathbf{e}_i$ the printer-dependent features, complementing a dataset-agnostic learning path (II) that learns from tonals the printer-independent features. The features learnt by the two paths are concatenated to be fed into a sub-network (III) that predicts the resulting spectral, CIELAB color and translucency α. On these predictions, we calculate (IV) the spectral loss $\mathbf{E}_{\text{ref}}$, the color loss $\mathbf{E}_{\text{color}}$, and the translucency loss $\mathbf{E}_{\text{tra}}$. Superscript (i,j) corresponds to the jth sample in the ith dataset $\mathcal{D}_i$. These losses are adjusted (V) by multiplying with our proposed adaptive loss weight c_i (VI) that depends on L^2 distance between the ith dataset $\mathcal{D}_i$'s embedding $\mathbf{e}_i$ and the targeted dataset $\mathcal{D}_\Gamma$'s embedding $\mathbf{e}_\Gamma$: the smaller this embedding distance, the bigger weight c_i is assigned to $\mathcal{D}_i$.

In contrast to RPDL where all samples are treated equally in the loss function, we treat samples discriminatively based on their underlying contributions to the characterization of the targeted Γth printer. We apply a function adaptation to assign bigger weights to samples of those datasets that are more similar to the targeted dataset than those that are less similar. Dataset similarity is evaluated via distance in dataset embedding space. Specifically, we adapt $\mathbf{E}_{\text{ref}}$, $\mathbf{E}_{\text{col}}$ and $\mathbf{E}_{\text{tra}}$ in Eq. (1) by applying a per-sample coefficient c_i (Fig. 2(V) and (VI)) as follows:

$$\mathbf{E}'_{\text{ref}} = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} \mathbf{E}_{\text{ref}}^{(i,j)} \tag{2}$$

$$\mathbf{E}'_{\text{col}} = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} \mathbf{E}_{\text{col}}^{(i,j)} \tag{3}$$

$$\mathbf{E}'_{\text{tra}} = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} c_i \sum_{j=1}^{n_i} \mathbf{E}_{\text{tra}}^{(i,j)} \tag{4}$$

where superscript (i,j) corresponds to the jth sample in the ith dataset $\mathcal{D}_i$, and c_i is a dataset-dependent weight shared by all the above three loss terms for samples within the ith dataset $\mathcal{D}_i$, with c_i defined as:

$$c_i = \exp(-\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2/\lambda) \tag{5}$$

where $\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2$ represents the L^2 distance between the dataset embeddings of datasets $\mathcal{D}_i$ and the targeted dataset $\mathcal{D}_\Gamma$ (Note the Γth dataset is from the targeted printer), and λ is a hyper-parameter shared across all printers.

Since $\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2 \in [0, \infty)$, we have $c_i \in (0, 1]$ with $c_i = 1$ for the targeted dataset itself. The more similar a dataset $\mathcal{D}_i$ is to the targeted $\mathcal{D}_\Gamma$, the smaller is the L^2 distance between their dataset embeddings (i.e. the smaller $\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2$), the bigger is the loss weight c_i for dataset $\mathcal{D}_i$. In this way, the contribution of characterization data from different printers to train the optical model for the targeted printer is considered based on their similarity to the targeted printer's characterization data.

The parameter λ in Eq. (5) is the only hyper-parameter to tune in the loss function. When the targeted Γth printer has sufficient samples in its dataset $\mathcal{D}_\Gamma$, samples from supporting printers might have adverse influences by over-constraining the neural network's capacity, in which case we need to reduce the influence from those supporting printers. On the other hand, when the targeted Γth printer has only scarce data, the neural network suffers from overfitting, in which case we need to increase the influence from those supporting printers that could provide extra constraints on the neural network. In the MPDL framework, the influence from those supporting printers is automatically adjusted via the tunable λ parameter: The larger λ, the larger is the influence from supporting printers, and vice versus. Note that λ is shared across all printers as a global controller. The automatic tuning process simply selects the $\lambda \in \{0.01, 0.1, 1, 10\}$ that minimizes the CIEDE2000 color difference on the validation data.

The final loss function in our MPDL framework is specified by replacing $\mathbf{E}_{\text{ref}}$, $\mathbf{E}_{\text{col}}$ and $\mathbf{E}_{\text{tra}}$ in Eq. (1) with $\mathbf{E}'_{\text{ref}}$, $\mathbf{E}'_{\text{col}}$ by $\mathbf{E}'_{\text{tra}}$ as defined in Eq. (2)–(4):

$$\mathbf{E} = \mathbf{E}'_{\text{col}} + a_1\mathbf{E}'_{\text{ref}} + a_2\mathbf{E}'_{\text{tra}} + a_3\mathbf{E}_{\text{mono\_LK}} + a_4\mathbf{E}_{\text{mono\_AT}} + a_5\mathbf{E}_{\text{lap}} \tag{6}$$

3.3. Remarks on the necessity of dataset embeddings

One benefit of using dataset embeddings in the proposed MPDL framework is that dataset embeddings allow a dataset-aware learning path (Section 3.1) to learn printer-dependent behaviors, complementing the dataset-agnostic learning path that learns only printer-independent behaviors (considering underlying printer similarities).

Another benefit of using dataset embeddings is that dataset embeddings allow adaptive loss weights (Section 3.2) for different datasets based on dataset embedding similarity between each supporting dataset and the targeted dataset. This adaptively adjusts the supporting datasets' impacts on the characterization of the targeted printer.

4. Contrastively learnt dataset embeddings

To obtain dataset embeddings $\{\mathbf{e}_i\}$ that are fed into the proposed MPDL framework, we propose a contrastive learning-based strategy. Contrastive learning has recently become a dominant technique due to its success in representation learning for natural language processing (NLP) and computer vision [31]. It typically learns to represent each input sample as a latent vector (also referred to as an embedding or representation) [32–34]. The learnt latent vector instead of the original sample is then used as an input of another learning task (referred to as a *downstream* task) e.g. an image classification task [35]. Contrastive learning requires none or just few data annotations (e.g. Images are annotated to belong to class 1 or class 2) which could be difficult or expensive in many applications, and this allows it to learn on a large amount of unlabeled data (e.g. numerous unlabeled images on Internet) to extract useful information for a downstream task whose training data is labeled but significantly smaller. We refer to Section 2.A of Supplement 1 for more details about the related works.

Similar to [32], our strategy to learn dataset embeddings comprises four major components:

1. A transformation module shown in Fig. 3 (I) that randomly transforms any dataset $\mathcal{D}_i$ into two vectors $\mathbf{v}_{i_1}$ and $\mathbf{v}_{i_2}$, that we call *views*. To derive a view from dataset $\mathcal{D}_i$, two successive steps are executed: 1) uniform-randomly sample a subset $\widetilde{\mathcal{D}}_{i_1} \subset \mathcal{D}_i$ of cardinality p (In this paper, we use $p = 100$); 2) rearrange elements of $\widetilde{\mathcal{D}}_{i_1}$ to a vector $\mathbf{v}_{i_1} = (\mathbf{t}_{i_1}^{(1)}, \mathbf{r}_{i_1}^{(1)}, \alpha_{i_1}^{(1)}, \ldots, \mathbf{t}_{i_1}^{(p)}, \mathbf{r}_{i_1}^{(p)}, \alpha_{i_1}^{(p)})$ where $\mathbf{t}_{i_1}^{(j)}$ is the tonal of jth sample of $\widetilde{\mathcal{D}}_{i_1}$, $\mathbf{r}_{i_1}^{(j)}$ is the corresponding spectral reflectance, and $\alpha_{i_1}^{(j)}$ is the corresponding translucency. Note that two views $\mathbf{v}_{i_1}$ and $\mathbf{v}_{i_2}$ derived from the same dataset can be correlated because the corresponding subsets $\widetilde{\mathcal{D}}_{i_1}, \widetilde{\mathcal{D}}_{i_2} \subset \mathcal{D}_i$ may have elements in common. A pair of views, $(\mathbf{v}_{i_1}, \mathbf{v}_{i_2})$, derived from the same dataset is called *positive pair*.

2. A neural network *encoder* f shown in Fig. 3(II) that produces intermediate representations from dataset views (i.e. from $\mathbf{v}_{i_1}$ and $\mathbf{v}_{i_2}$). For simplicity, we adopt the encoder architecture from [34] that is a fully-connected network comprising 4 hidden layers each with K=256 neurons.

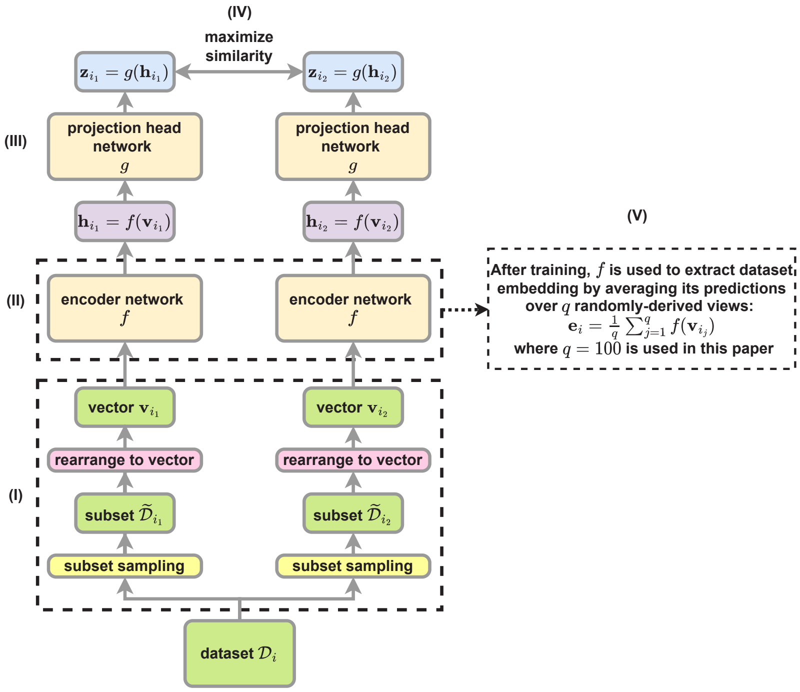
(IV)
maximize similarity
$\mathbf{z}_{i_1} = g(\mathbf{h}_{i_1})$
$\mathbf{z}_{i_2} = g(\mathbf{h}_{i_2})$
(III)
projection head network g
projection head network g
$\mathbf{h}_{i_1} = f(\mathbf{v}_{i_1})$
$\mathbf{h}_{i_2} = f(\mathbf{v}_{i_2})$
(V)
(II)
encoder network f
encoder network f
After training, f is used to extract dataset embedding by averaging its predictions over q randomly-derived views:
$\mathbf{e}_i = \frac{1}{q}\sum_{j=1}^{q} f(\mathbf{v}_{i_j})$
where $q = 100$ is used in this paper
vector $\mathbf{v}_{i_1}$
vector $\mathbf{v}_{i_2}$
rearrange to vector
rearrange to vector
(I)
subset $\widetilde{\mathcal{D}}_{i_1}$
subset $\widetilde{\mathcal{D}}_{i_2}$
subset sampling
subset sampling
dataset $\mathcal{D}_i$

Fig. 3. Overview of a proposed strategy to learn dataset embeddings. Each dataset $\mathcal{D}_i$ is processed by a transform module (I) that consists of a subset sampling step resulting in 2 sub-datasets i.e. $\widetilde{\mathcal{D}}_{i_1}$ and $\widetilde{\mathcal{D}}_{i_2}$ and a successive step that rearranges elements of each subset into a vector views $\mathbf{v}_{i_1}$ or $\mathbf{v}_{i_2}$. The 2 views are fed into (II) a neural network *encoder* f to produce intermediate representations $\mathbf{h}_{i_1}$ and $\mathbf{h}_{i_2}$. The intermediate representations $\mathbf{h}$ is further mapped by (III) a small neural network *projection head* g to high-level features $\mathbf{z}_{i_1}$ and $\mathbf{z}_{i_2}$. A *contrastive loss function* (IV) is calculated on $\mathbf{z}$ to encourage the 2 derived views from the same dataset to be similar in $\mathbf{z}$ feature space. After training, dataset embedding (V) is obtained by averaging $\mathbf{h}$ over multiple randomly-derived views.

3. A small neural network g, called *projection head*, shown in Fig. 3 (III) maps dataset embeddings to higher level features i.e. $\mathbf{z}_i = g(\mathbf{h}_i)$ on which a contrastive loss (see below) is calculated. We adopt the projection head architecture from [34] that is a fully-connected network comprising 2 hidden layers each with 256 neurons. With the loss calculated on $\mathbf{z}$, [32] and [36] observed performance gains by using $\mathbf{h}$ rather than $\mathbf{z}$ as learnt representation for downstream tasks, and based on experiments they showed that $\mathbf{h}$ contains more information for downstream tasks than the condensed $\mathbf{z}$.

4. A *contrastive loss function* used in Fig. 3(IV) encourages each positive pair of views (i.e. derived from the same dataset) to be similar in $\mathbf{z}$ feature space, and each negative pair of views (i.e. derived from 2 different datasets) to be dissimilar in $\mathbf{z}$ feature space. Similar to [32–34], we adopt the *InfoNCE* [37] loss that is commonly used for contrastive learning, and we calculate it on each positive pair of samples (i, j):

$$\mathcal{L}(i,j) = -\log \frac{\exp(s_{i,j})}{\sum_{k\in\mathbf{V}|k\neq i}\exp(s_{i,k})} \tag{7}$$

where $s_{i,j} = \mathbf{z}_i^T\mathbf{z}_j/(\|\mathbf{z}_i\|_2\|\mathbf{z}_j\|_2)$ measures the cosine similarity between $\mathbf{z}_i$ and $\mathbf{z}_j$, $\mathbf{V} = \{1_1, 1_2, \ldots, m_1, m_2\}$ is the index set corresponding to the $2m$ views from the m datasets. Note that negative pairs of views are used in the denominator in Eq. (7), i.e. one view from each considered dataset is compared with view i.

Algorithm 1. Contrastive learning-based algorithm for learning dataset embeddings.

```
Algorithm 1 Contrastive learning-based algorithm for learning dataset embeddings
input: $m$ datasets $\{\mathcal{D}_i\}_{i=1}^m$, encoder network $f$, projection head network $g$, constant $p$
for each training step do
    for all $i \in \{1, ..., m\}$ do
        uniformly sample 2 subsets $\widetilde{\mathcal{D}}_{i_1}, \widetilde{\mathcal{D}}_{i_2} \subset \mathcal{D}_i$ of cardinality $p$
        rearrange elements of $\widetilde{\mathcal{D}}_{i_1}$ into a vector $\mathbf{v}_{i_1} = (\mathbf{t}_{i_1}^{(1)}, \mathbf{r}_{i_1}^{(1)}, \alpha_{i_1}^{(1)}, ..., \mathbf{t}_{i_1}^{(p)}, \mathbf{r}_{i_1}^{(p)}, \alpha_{i_1}^{(p)})$.
    Similarly rearrange elements of $\widetilde{\mathcal{D}}_{i_2}$ into a vector $\mathbf{v}_{i_2}$    // derive 2 views from each dataset
        let $\mathbf{z}_{i_1} = g(f(\mathbf{v}_{i_1})), \mathbf{z}_{i_2} = g(f(\mathbf{v}_{i_2}))$
    end for
    let $\mathbf{V} = \{1_1, 1_2, ..., m_1, m_2\}$  // the index set corresponding to $2m$ views from $m$ datasets
    for all $i, j \in \mathbf{V}$ do
        let $s_{i,j} = \mathbf{z}_i^T\mathbf{z}_j/(\|\mathbf{z}_i\|_2\|\mathbf{z}_j\|_2)$    // dataset-level pairwise similarity
    end for
    define $\mathcal{L}(i,j) = -\log \frac{\exp(s_{i,j})}{\sum_{k\in\mathbf{V}|k\neq i}\exp(s_{i,k})}$
    let $\mathcal{L} = \frac{1}{m}\sum_{k=1}^m[\mathcal{L}(k_1,k_2) + \mathcal{L}(k_2,k_1)]/2$    // average loss on positive pair $(k_1, k_2)$
    update networks $f$ and $g$ to minimize $\mathcal{L}$
end for
return encoder network $f$, discard network $g$
```

The networks f and g are trained jointly according to Algorithm 1. Note the 2 derived views from each dataset are re-sampled at every training iteration in order to well cover the dataset. After training, for each dataset $\mathcal{D}_i$, we use f to obtain the dataset embedding by averaging f's output i.e. $\mathbf{h}$ representations predicted on q randomly-derived views of $\mathcal{D}_i$. We use $q = 100$ in this paper. Specifically, $\mathcal{D}_i$'s dataset embedding $\mathbf{e}_i$ is obtained by $\mathbf{e}_i = (\sum_{j=1}^q \mathbf{h}_{i_j})/q = (\sum_{j=1}^q f(\mathbf{v}_{i_j}))/q$ where $\mathbf{v}_{i_j}$ is the jth of q views derived from dataset $\mathcal{D}_i$. The resulting dataset embeddings $\{\mathbf{e}_i\}$ are used in the proposed MPDL framework described in Section 3. In Section 2.B of Supplement 1, we further discuss the necessity and advantages of this averaging way of calculating dataset embeddings.

5. Experiments

5.1. Datasets

We use all four datasets described in [27,28] 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)): Two datasets to characterize Stratasys J750 printers and two datasets to characterize Mimaki 3DUJ-553 printers, respectively. The Stratasys printers use *VeroCyan-V, VeroMagenta-V, VeroYellow-V, VeroBlack, VeroPureWhite*, and *VeroClear* materials and the Mimaki printers use *MH100-C-BA, MH100-M-BA, MH100-Y-BA, MH100-K-BA, MH100-W-BD*, and *MH100-CL-BD* materials. The datasets consist of reflectance and α-measurements of printed flat targets with known tonal values, except for those datasets where all samples are opaque and thus α-measurements were not collected and not available for experiments. We denote the two Stratasys datasets as *Stratasys 1* and *Stratasys 2*, and the two Mimaki datasets as *Mimaki 1* and *Mimaki 2*. We refer to [27] for details on the set of sampled tonal values and the measurement setup. In addition, we collected four more datasets from Stratasys printers using the same measurement setup:

Stratasys 3: The data comes from a printing system similar to Stratasys 1 except that the rigid *VeroCyan-V, VeroMagenta-V, VeroYellow-V, VeroPureWhite* materials have been replaced by flexible *AgilusCyan, AgilusMagenta, AgilusYellow, AgilusWhite* materials. The dataset consists of a regular grid $\{0, 63.75, 127.5, 191.25, 255\}^4 \subset$ CMYK of tonal values, and 1000 random CMYK-samples. All samples are opaque, i.e. $\mathrm{T} = 0$. In total there are $5^4 + 1000 = 1625$ samples.

Stratasys 4: The data comes from a different instance of a printing system similar to Stratasys 1. It consists of 450 random CMYKT-samples and 100 CMYKT-samples randomly selected from *Stratasys 1* (i.e. The two corresponding printers printed these 100 samples using the same material arrangement controlled by tonal values, resulting in 100 printed samples from each printer, which can be used to investigate inter-printer variability). In total there are 450 + 100 = 550 samples.

Stratasys 5: The data consists of the 450 random CMYKT-samples from *Stratasys 4* printed by a different instance of the same printing system, to investigate inter-printer variability.

Stratasys 6: The samples have been printed on a Stratasys *J850* printer with *VeroCyan-V, VeroMagenta-V, VeroYellow-V, UltraBlack, UltraWhite, VeroClear* and *UltraClear* materials. *UltraClear* and *VeroClear* are both controlled by the same tonal value T whereas UltraClear is used in the core of the prints and VeroClear close to the surface. The dataset consists of a regular grid $\{0, 127.5, 255\}^5 \subset$ CMYKT of tonal values, 1757 random CMYKT-samples, 1500 random opaque CMYK-samples with $\mathrm{T} = 0$, 500 random CMYK-samples with $\mathrm{T} = 255$, 187 light-color opaque CMYK-samples, and 113 random CMYKT-samples from *Stratasys 2*. In total there are $3^5 + 1757 + 1500 + 500 + 187 + 113 = 4300$ samples.

In summary, the characterization data covers a wide range of different printing systems (models and materials) and includes also different instances of the same printing system allowing to investigate model performance on inter-printer dissimilarities .

5.2. Computing and evaluating predictions

Random samples from the dataset belonging to the targeted printer 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. The training set from the targeted printer, as well as all samples from supporting datasets, are used for training. Validation data and test data are only from the targeted printer.

Similarly to the PDL and RPDL models [27,28], non-test samples of the targeted printer are uniform-randomly split into validation samples and training samples as per the ratio being validation:training=10%:90%. This random validation-training splitting is performed 10 times independently with different randomizations e.g. via different random seeds across different splitting runs. For each random splitting, a MPDL model is trained on all training samples (i.e. the training samples from the targeted printer, plus all samples from all other printers), and the model's hyperparameters are fit based on the validation samples from the targeted printer. The 10 different validation-training splittings result in 10 slightly differently trained models. After training, we averaged the predictions computed by these 10 models. Note that the union of training and validation data is exactly the same for the 10 models, with the same test data always unseen.

The contrastive learning neural network is trained on the same union of training and validation data as the neural network in the MPDL framework.

5.3. Environment and model training setup

All neural networks are implemented with TensorFlow 2.2.0, and trained on an NVIDIA GeForce RTX 3090 GPU. Training a MPDL neural network takes approx. 300s for 282 training samples of *Stratasys 1* as targeted printer and a total of 19317 samples from supporting printers, i.e. totally 282+19317=19599 training samples. We observe a similar runtime for other targeted printers if a similar number of training samples are used. Training a contrastive learning neural network takes approx. 30s.

For the MPDL neural network, Adam optimizer with an initial learning rate of 0.003 is used. The same learning rate decay as in [28] is used. For the contrastive learning neural network, Adam optimizer with a fixed learning rate 0.0003 is used.

5.4. Results

In Fig. 4, we compare MPDL with the state-of-the-art RPDL model [28] in terms of prediction accuracy. All results are evaluated on the test data of the targeted printer. Models trained on the entire union of training and validation data are indicated with the term *Big Data*. To investigate the performance of the models on much smaller data, we also trained some models with only 10% of the union of training and validation data. These models are indicated with the term *Small Data*. To show how good RPDL generalizes across printers, we show also the best results of the RPDL model on the targeted printer while trained on another printer's dataset (The targeted printer's data was not used for training). These results are indicated by *best reference model*.

The results in Fig. 4 show:

1. When directly reusing the best reference model, the mean color prediction error on the targeted dataset is high, ranging from CIEDE2000 = 2 to 18. One example is *Stratasys 6*, for which the best reference model is trained on *Stratasys 4* data yielding CIEDE2000 around 5. Another example is *Stratasys 3*, for which the best reference model has a very poor average CIEDE2000 prediction error of 10. These two printing systems use very different materials than all the other printing systems, as described in Section 5.1. A color difference of CIEDE2000 > 5 is not acceptable for most color critical applications. This shows the poor model generalization performance across different printing systems.

   Characterization datasets from different instances of the same printing system (*Stratasys 1-2* and *4-5*) indicate also a noticeable inter-printer variability because prediction errors from the best reference model are noticeably larger than from RPDL trained on the full targeted training dataset. The best reference model's prediction errors range from CIEDE2000 = 2

Optics EXPRESS

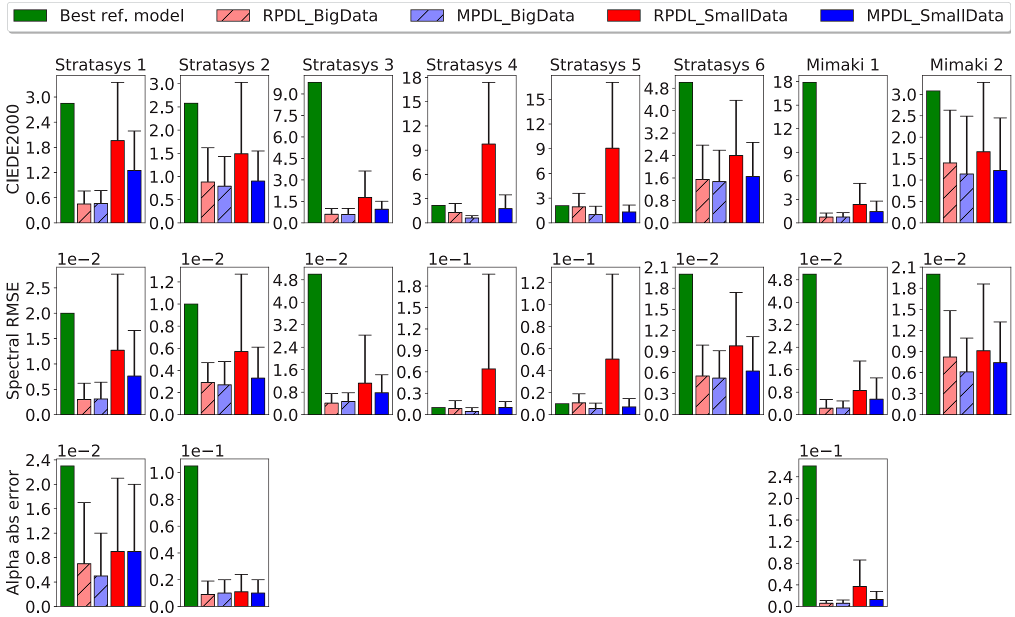

Fig. 4. Accuracy comparison between RPDL and MPDL. Each bar shows the average prediction error, and the whisker above the bar shows the 90-th percentile of that prediction error. For the best reference model, only the average error is shown. Note translucency α measurements are not available for Stratasys 3-6 and Mimaki 2, thus there are no translucency α plots for these datasets.

to 3, which might not be acceptable for color critical applications, such as for prosthetic eyes or dental restorations. This shows also the limits of using an universal color profile for different instance of a printing systems.

2. MPDL outperforms the state-of-the-art RPDL model learned on just the targeted characterization data by a large margin especially for small data: For small data of *Stratasys 5*, MPDL improves color accuracy by up to 6.8 times. For *Stratasys 2*, *Stratasys 5*, and *Mimaki 2*, MPDL trained on small data leads to an accuracy comparable with RPDL trained on *big* data, meaning 10X data reduction for a similar performance. This verifies the data efficiency of MPDL employing supporting data from other printers.

3. Surprisingly and interestingly, despite *very* poor performance of the best reference models due to the differences among printers, adding data from supporting printers into MPDL can significantly improve accuracy for the targeted printer: For *Mimaki 1*, the best reference model has a very high prediction error of CIEDE2000=18, indicating that the targeted printer is very different to the other printers. The RPDL model on small data has an average prediction error of CIEDE2000=2.3, but adding supporting datasets into MPDL significantly reduces the prediction error to CIEDE2000=1.4 on small data, i.e. by about 40%. Even if printing systems are very different their characterization data possess some underlying similarities, so it is beneficial to let the model learn printer-independent general features on multiple datasets simultaneously, because the supporting datasets provide extra constraints as regularization to improve the model's generalization performance on the targeted dataset. This verifies the effectiveness of MPDL employing supporting datasets from other printers. We speculate that underlying information shared by datasets from *very* different printers can be exploited by the MPDL framework: Potentially it is related

to the physics of light transport or similar strategies for halftoning, i.e. statistically similar droplet positioning that ideally has blue-noise characteristic [38]. This is an interesting research question that we cannot answer in this paper.

4. The second row of subplots shows that spectral prediction errors (Root Mean Square Error, or RMSE) have a similar trend as color prediction error (CIEDE2000) and that we can draw the same conclusions as in 1-3.

5. The third row of subplots shows that the accuracy improvement on translucency α predictions is not as large as for color or spectral predictions, even though translucency data is available only in three datasets. Nevertheless, translucency α accuracy is already much lower than the Just-Noticeable-Difference (JND) of approx $\Delta\alpha = 0.1$ [29]. This means it is sufficient for most applications.

5.5. Visualization of spectral predictions

In Fig. 5, we compare the proposed MPDL with RPDL and the best reference model by visualizing their spectral predictions if trained on small data (i.e. only 10% training data from the targeted printer is used for training MPDL and RPDL). The sRGB color of each spectral prediction is also shown. The figure shows that MPDL's predictions match the groundtruth significantly better than the two counterparts, e.g. in the first subplot of the first row (Stratasys 3) where the two counterparts' predicted spectrals and colors are both way off from the groundtruth. This indicates a much higher accuracy from MPDL.

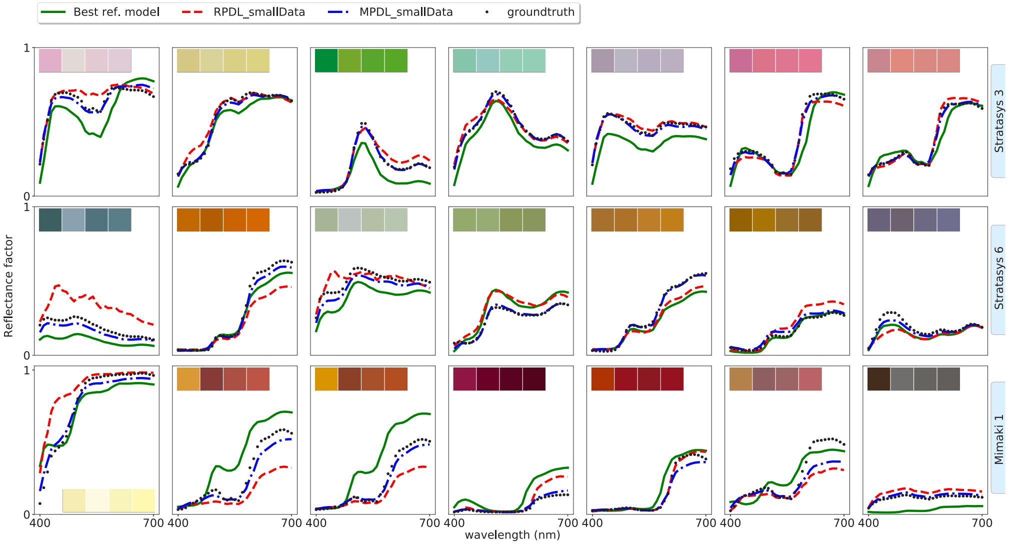

Fig. 5. Spectral predictions by different models. The 3 rows correspond to datasets *Stratasys 3*, *Stratasys 6*, and *Mimaki 1*, respectively. The 7 subplots in each row correspond to the 7 test samples on which RPDL and MPDL have the largest root mean square difference. The sRGB color of each spectral prediction is also shown at the upper-left corner of the corresponding subplot, in the order of the best reference model, RPDL with small data (i.e. 10% training data), MPDL with small data, and groundtruth.

5.6. MPDL captures post-processing influence

We compare MPDL trained on small data for *Stratasys 6* for samples on which a plastic care treatment was applied as post-processing, with the best reference RPDL model trained on all available data of a post-processing-free version of *Stratasys 6*. Fig. 6 shows histograms of the spectral prediction RMSE of these two models. It shows that even with only 10% data, MPDL leads to lower spectral prediction errors on dataset *Stratasys 6* than reusing the best reference model trained on a very similar *full*-size dataset (i.e. the post-processing-free version of dataset *Stratasys 6*).

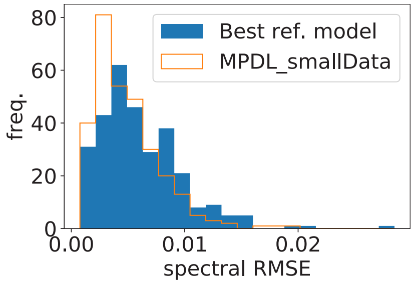

Fig. 6. Spectral RMSE histogram

6. Limitations and future work

We see the dataset embedding-based learning framework MPDL as our major contribution, and regard the contrastive learning-based strategy of learning dataset embeddings a minor contribution. As a replaceable component, the contrastive learning-based strategy of learning dataset embeddings is not fully exploited, with most settings (e.g. the projection head neural network g during training) adopted from related works that are optimized for different learning tasks e.g. image classification. For future work, we plan to explore the use of contrastive learning-based strategies for exploiting dataset embeddings, as well as alternative methods of learning dataset embeddings optimized for optical printer modelling.

7. Conclusion

In this paper, we propose a methodology to improve the prediction accuracy and data efficiency of an optical printer model for characterizing a targeted printing system, by exploiting characterization data from other printing systems. For that, we proposed a dataset embedding-based Multi-Printer Deep Learning (or MPDL) framework with a 2-path feature learning strategy that not only learns printer-independent features shared across multiple printers, but also learns from dataset embeddings the printer-dependent features. The MPDL framework also uses dataset embedding-based adaptive loss weights to balance the supporting datasets' respective influences on the model. We provided a learning strategy for training the model and fitting its hyperparameters. On the other hand, to provide dataset embeddings that are required by the MPDL framework, we propose a contrastive leaning-based approach to learn dataset embeddings.

Experiments on datasets from eight state-of-the-art multi-material 3D printing systems show that the MPDL framework can drastically reduce the number of required samples that have to be printed, measured and postprocessed to characterize a targeted printing system, for achieving an application-specific prediction accuracy. For some printers, the MPDL framework requires 10% of the samples to achieve a similar accuracy as the state-of-the-art RPDL model [28]. This makes it economically possible to frequently characterize machineries of 3D printers to achieve more consistent output across different printers with higher accuracy over time, which is crucial for color- and translucency-critical individualized mass production, such as 3D printed prosthetic eyes or dental restorations. A smaller number of characterization samples is also very beneficial for characterizing desktop 3D printing systems that possess a very small build space and thus require more separate print runs than other printing systems. Note each separate print run requires additional efforts of cleaning print heads and the print tray. The data efficiency improvement from MPDL reduces the number of separate print runs and thus the overall characterization efforts.

Although conditioned on dataset embeddings, the MPDL framework is general and not limited to any a certain way of obtaining dataset embeddings.

Appendix A. Symbol lookup table

In Table 1 we summarize symbols in the alphabetical order, with Greek symbols at the end.

Table 1. Symbol lookup table

α	A translucency scalar. $\alpha \in \mathcal{A}$
$\alpha_i^{(j)}$	The translucency of the jth sample of the ith dataset.
c_i	The loss weight for the ith dataset. $c_i = \exp(-\|\mathbf{e}_i - \mathbf{e}_\Gamma\|_2/\lambda)$
$\mathbf{e}_i$	The dataset embedding of the ith dataset. $\mathbf{e}_i \in \mathcal{E}$
m	The number of datasets used in the proposed framework
n_i	The cardinality of the ith dataset, i.e. $n_i = \|\mathcal{D}_i\|$
q	The number of randomly-derived views for calculating dataset embedding
$\mathbf{r}$	A spectral vector. $\mathbf{r} \in \mathcal{S}$
$\mathbf{r}_i^{(j)}$	The spectral of the jth sample of the ith dataset.
$\mathbf{t}$	A tonal vector. $\mathbf{t} \in \mathcal{T}$
$\mathbf{t}_i^{(j)}$	The tonal of the jth sample of the ith dataset.
$\mathbf{v}_{i_j}$	The jth derived view of $\mathcal{D}_i$
$\mathcal{A}$	Translucency space. $\mathcal{A} = [0, 1]$
$\mathcal{D}_i$	The dataset collected from the ith printer. $\mathcal{D}_i = \{(\mathbf{t}_i^{(j)}, \mathbf{r}_i^{(j)}, \alpha_i^{(j)})\}_{j=1}^{n_i}$
$\widetilde{\mathcal{D}}_{i_j}$	The jth random subset of $\mathcal{D}_i$. $\widetilde{\mathcal{D}}_{i_j} \subset \mathcal{D}_i$
$\mathcal{E}$	Dataset embedding space. $\mathcal{E} = \mathbb{R}^K$ (K=256 in the paper)
M	The dimensions of tonal space.
N	The dimensions of Spectral space, i.e. the number of considered wavelengths.
$\mathcal{S}$	Spectral space. $\mathcal{S} = [0, 1]^N$
$\mathcal{T}$	Tonal space. $\mathcal{T} = [0, 1]^M$
Γ	The dataset index of the targeted printer. $1 \leq \Gamma \leq m$
λ	A hyperparameter for calculating c_i

Appendix B. Loss terms used by RPDL [28]

The loss function of the PDL [27] consists of three parts: 1) the spectral root-mean-square error (RMSE) $\mathbf{E}_{\text{ref}}$, 2) the CIEDE2000 color error $\mathbf{E}_{\text{col}}$ computed for specific viewing conditions (illuminant, observer), and 3) the α-based translucency error $\mathbf{E}_{\text{tra}}$, defined respectively as follows:

$$\mathbf{E}_{\text{ref}} = \frac{1}{n}\sum_{j=1}^{n}\sqrt{\frac{1}{N}\|\hat{\mathbf{r}}^{(j)} - \mathbf{r}^{(j)}\|_2^2} \tag{8}$$

$$\mathbf{E}_{\text{col}} = \frac{1}{n}\sum_{j=1}^{n}\Delta E_{00}(\mathbf{LAB}(\hat{\mathbf{r}}^{(j)}), \mathbf{LAB}(\mathbf{r}^{(j)})) \tag{9}$$

$$\mathbf{E}_{\text{tra}} = \frac{1}{n}\sum_{j=1}^{n}|\hat{\alpha}^{(j)} - \alpha^{(j)}|. \tag{10}$$

where $\hat{\mathbf{r}}^{(j)}, \mathbf{r}^{(j)} \in \mathcal{S}$ are the predicted and the measured reflectances respectively for the j-th sample within a minibatch of n samples, $\mathbf{LAB} : \mathcal{S} \mapsto$ CIELAB is the function that computes CIELAB values from reflectances assuming specified viewing conditions, ΔE_{00} means CIEDE2000 color difference, $\hat{\alpha}^{(j)}, \alpha^{(j)} \in \mathcal{A}$ are the predicted and the measured translucency α-values [29] respectively for the j-th sample.

Adopting the above 3 loss terms, RPDL [28] introduced 3 more loss terms to improve physical robustness and plausibility: 1) monotonicity loss $\mathbf{E}_{\text{mono\_LK}}$ for lightness CIE-L^* w.r.t. black material K, 2) monotonicity loss $\mathbf{E}_{\text{mono\_AT}}$ for translucency α w.r.t. transparent material T, and 3) Laplacian smoothness loss $\mathbf{E}_{\text{lap}}$, defined respectively as follows:

$$\mathbf{E}_{\text{mono\_LK}}(\mathcal{M}_{\text{LK}}) = \frac{1}{|\mathcal{M}_{\text{LK}}|} \sum_{\tau \in \mathcal{M}_{\text{LK}}} \left. \frac{\partial \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial \mathbf{t}_{\text{K}}} \right|_{\mathbf{t}=\tau} \tag{11}$$

$$\mathbf{E}_{\text{mono\_AT}}(\mathcal{M}_{\text{AT}}) = \frac{1}{|\mathcal{M}_{\text{AT}}|} \sum_{\tau \in \mathcal{M}_{\text{AT}}} \left. \frac{\partial \mathbf{A}(\mathbf{PDL}(\mathbf{t}))}{\partial \mathbf{t}_{\text{T}}} \right|_{\mathbf{t}=\tau} \tag{12}$$

$$\mathbf{E}_{\text{lap}}(\mathcal{M}) = \frac{1}{M|\mathcal{M}|} \sum_{\tau \in \mathcal{M}} \sum_{i=1}^{M} \log \left(\left| \left. \frac{\partial^2 \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial \mathbf{t}_i^2} \right|_{\mathbf{t}=\tau} \right| + 1 \right). \tag{13}$$

where $\mathbf{t} \in \mathcal{T} = [0, 1]^M$, $\mathbf{t}_{\text{K}}$ and $\mathbf{t}_{\text{T}}$ are the element of $\mathbf{t}$ corresponding to the black and transparent materials respectively, $\mathbf{t}_i$ is the ith element of $\mathbf{t}$, $\mathbf{L} : \mathcal{S} \times \mathcal{A} \mapsto [0, 100]$ extracts lightness CIE-L^* from PDL model predictions, i.e. it uses the predicted reflectance and computes lightness for a given viewing condition (illuminant, observer), $\mathbf{A} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{A}$ extracts translucency α-values from PDL model predictions, $\mathcal{M} \subset \mathcal{T}$ is a set of tonals randomly selected from tonal space $\mathcal{T}$ at each training iteration, $\mathcal{M}_{\text{LK}} \subset \mathcal{M}$ is $\mathcal{M}$'s subset containing positive derivatives of CIE-L^* w.r.t. the black material, i.e.

$$\mathcal{M}_{\text{LK}} = \left\{ \tau \in \mathcal{M} \mid \left. \frac{\partial \mathbf{L}(\mathbf{PDL}(\mathbf{t}))}{\partial \mathbf{t}_{\text{K}}} \right|_{\mathbf{t}=\tau} > 0 \right\} \tag{14}$$

and similarly, $\mathcal{M}_{\text{AT}} \subset \mathcal{M}$ is $\mathcal{M}$'s subset containing positive derivatives of α w.r.t. the transparent material:

$$\mathcal{M}_{\text{AT}} = \left\{ \tau \in \mathcal{M} \mid \left. \frac{\partial \mathbf{A}(\mathbf{PDL}(\mathbf{t}))}{\partial \mathbf{t}_{\text{T}}} \right|_{\mathbf{t}=\tau} > 0 \right\} \tag{15}$$

Note $\mathbf{E}_{\text{mono\_LK}}$, $\mathbf{E}_{\text{mono\_AT}}$ and $\mathbf{E}_{\text{lap}}$ are heuristic losses calculated on arbitrary sampled tonals without measured "groundtruth", hence $\mathcal{M}$ is re-sampled from tonal space $\mathcal{T}$ at each training iteration to better cover the whole space. In the paper, we use CIED50 as illuminant and the CIE1936 color matching functions as observer. Readers are referred to RPDL [28] for more details about the above equations.

Funding. H2020 Marie Skłodowska-Curie Actions (813170).

Disclosures. The authors declare no conflicts of interest.

Data availability. Datasets of Stratasys 1 and 2, Mimaki 1 and 2 are available via [27] and [28]. Other datasets are 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. Murray, "Monochrome reproduction in photoengraving," J. Franklin Inst. **221**(6), 721–744 (1936).
2. H. E. J. Neugebauer, "The Theoretical Basis of Multicolor Letterpress Printing (Translated D. Wyble and A. Kraushaar)," Color Res. Appl. **30**, 322–331 (2005).
3. R. Rolleston and R. Balasubramanian, "Accuracy of Various Types of Neugebauer Model," in *IS&T/SID*, (Scottsdale Ariz., 1993), pp. 32–36.
4. J. A. C. Yule and W. J. Nielsen, "The penetration of light into paper and its effect on halftone reproduction," in *Tech. Assn. Graphic Arts*, vol. 4 (1951), pp. 65–76.

5. J. A. C. Yule and R. S. Colt, "Colorimetric Investigations in Multicolor Printing," in *TAGA Proceedings*, (1951), pp. 77–82.
6. J. Viggiano, "Modeling the Color of Multi-color Halftones," in *TAGA Proceedings*, (1990), pp. 44–62.
7. R. Hersch, P. Emmel, F. Collaud, and F. Crété, "Spectral reflection and dot surface prediction models for color halftone prints," J. Elect. Imaging **14**(3), 033001 (2005).
8. R. Hersch and F. Crété, "Improving the yule-nielsen modified spectral neugebauer model by dot surface coverages depending on the ink superposition conditions," Proc. SPIE **5667**, 434–445 (2005).
9. 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.
10. F. Clapper and J. Yule, "Reproduction of color with halftone images," in *Proc. Seventh Ann. Tech. Meet. TAGA*, (1955), pp. 1–14.
11. 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).
12. R. Hersch, F. Collaud, and P. Emmel, "Reproducing color images with embedded metallic patterns," ACM Trans. Graph. **22**(3), 427–434 (2003).
13. G. Rogers, "A generalized clapper–yule model of halftone reflectance," Color Res. Appl. **25**, 402–407 (2000).
14. M. Hébert and R. D. Hersch, "Review of spectral reflectance models for halftone prints: principles, calibration, and prediction accuracy," Color Res. Appl. **40**, 383–397 (2015).
15. A. U. Agar and J. P. Allebach, "An iterative cellular ynsn method for color printer characterization," in *IS&T/SID*, (Scottsdale Ariz., 1998), pp. 197–200.
16. V. Babaei and R. D. Hersch, "*n*-ink printer characterization with barycentric subdivision," IEEE Trans. on Image Process. **25**(7), 3023–3031 (2016).
17. P. Kubelka and F. Munk, "Ein Beitrag zur Optik der Farbanstriche," Zeitschrift für Technische Physik **12**, 593–601 (1931).
18. J. Saunderson, "Calculation of the color of pigmented plastics," J. Opt. Soc. Am. **32**(12), 727–729 (1942).
19. 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.
20. 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.
21. L. Simonot, R. D. Hersch, M. Hébert, and S. Mazauric, "Multilayer four-flux matrix model accounting for directional-diffuse light transfers," Appl. Opt. **55**(1), 27 (2016).
22. 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**(6), 065004 (2018).
23. 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.
24. 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.
25. 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**, 132–175 (2002).
26. 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 Tran. Graph. **37**(6), 1–15 (2018).
27. D. Chen and P. Urban, "Deep learning models for optically characterizing 3d printers," Opt. Express **29**(2), 615–631 (2021).
28. D. Chen and P. Urban, "Inducing robustness and plausibility in deep learning optical 3d printer models," Opt. Express **30**(11), 18119–18133 (2022).
29. 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).
30. D. R. Wyble and R. S. Berns, "A Critical Review of Spectral Models Applied to Binary Color Printing," Color Res. Appl. **25**, 4–19 (2000).
31. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 1 (MIT press Cambridge, 2016).
32. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research* H. D. III and A. Singh, eds. (PMLR, 2020), pp. 1597–1607.
33. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020).
34. D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "Scarf: Self-supervised contrastive learning using random feature corruption," in *International Conference on Learning Representations*, (2022).
35. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (2009), pp. 248–255.
36. X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," arXiv, arXiv:2003.04297 (2020).

37. A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv, arXiv:1807.03748 (2018).

38. D. L. Lau and G. R. Arce, *Modern Digital Halftoning* (CRC Press, 2001).