

RESEARCH ARTICLE



A machine learning approach to color space Euclidization

Lia Ahrens¹ | Julian Ahrens¹ | Hans D. Schotten^{1,2}

¹Department of Intelligent Networks, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

²Department of Electrical and Computer Engineering, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU), Kaiserslautern, Germany

Correspondence

Lia Ahrens, Department of Intelligent Networks, German Research Center for Artificial Intelligence (DFKI), Trippstadter Str. 122, 67663 Kaiserslautern, Germany. Email: lia.ahrens@dfki.de

[Correction added on 11 October 2023, after first online publication: The JEL classification was removed from the abstract.]

Abstract

In this work, a machine learning methodology is proposed for the issue of color space Euclidization. Given a color difference formula as reference distance law, the Euclidization task consists in finding an injective transformation from the original color space into a real vector space and the corresponding inverse transformation, such that the Euclidean distances in the embedded color space align with the reference color distances. For this, artificial neural networks are devised as function approximators for the color space transformations being sought. Training these neural networks is accomplished through unsupervised learning, making use of random sampling and gradient descent. As key disagreement measure, either the (symmetric) relative isometric disagreement or the standardized residual sum of squares (STRESS) index is considered at a time and incorporated as part of the optimization criterion into the objective function. Comparative evaluation is carried out on well-established color distance laws, including the CIELAB-based DE2000 color difference formula. The evaluation results indicate significant performance advantages of the proposed approach over previous contributions.

KEYWORDS

artificial neural networks, color space Euclidization, DE2000 color difference formula, machine learning, perceptually uniform color space

1 | INTRODUCTION

The derivation of perceptually uniform color spaces is considered one of the key issues in colorimetry. Back in 1943, based on the experimental data collected by MacAdam,¹ Siberstein² constructed a curvilinear coordinate system on the CIE 1931 xy chromaticity diagram whose coordinate curves are parallel to the principal axes of the MacAdam ellipses. He also sketched a method for constructing a surface embedded in three-dimensional Euclidean space from this coordinate system, such that perceived color distances are faithfully represented by geodesics (shortest paths) on the surface. A key observation

was that the resulting surface could never be flat and that its curvature would be of quite noticeable proportion. In the following decades, extensive research has been conducted on relating human visual color perception to a specified color coordinate system, which led to color models such as the CIELUV and CIELAB color spaces,³ as well as the perceptually uniform color spaces (UCS) based on the CIECAM02⁴ and CIECAM16⁵ color appearance models. Accompanying the development of such models, refined color difference formulas were derived, augmenting the original color space and thereby allowing for more accurate predictions of visually perceived color differences. As a well-known instance of this, a series of color difference

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2023 The Authors. Color Research and Application published by Wiley Periodicals LLC.

formulas based on the CIELAB color space came into existence, including the CMC,⁶ BFD,⁷ CIE94,⁸ DIN99d,⁹ and DE2000¹⁰ color difference formulas, with the last one currently being recommended as standard by the International Commission on Illumination (CIE) for small color differences below 5 CIELAB-units (cf. Reference 11). One major concern over these performance-enhancing color difference formulas consists in the non-Euclidean nature of the corresponding metric spaces, which is rather undesirable in practical applications. If the original color space endowed with the considered color difference formula (so-called color difference system) could be embedded isometrically into a Euclidean space, a new perceptually uniform color space would emerge. At this point, it should be noted that, in a sense, the work by Silberstein² already represents an early instance of an attempt at finding such an isometric embedding, albeit directly using experimental data in absence of a reference color difference formula.

In general, in order for an isometric embedding into a Euclidean space (so-called Euclidization) to exist, the color difference formula has to satisfy the following conditions:

1. By definition, an isometric embedding is a map between two metric spaces that preserves distances. Thus, for an isometric embedding to exist, it is necessary that the color difference formula satisfies the defining properties of a metric, that is, positivity, symmetry in its two arguments, and the triangle inequality. (This condition is not always fulfilled for existing color difference formulas. For instance, CMC and CIE94 are asymmetric in their two arguments; when being Euclidized, this may be compensated for through symmetrization such as in Reference 12.)
2. While the original color space can always be trivially embedded into a metric space through the identity mapping once the underlying color difference formula indeed defines a metric, limiting the choice of codomain to a Euclidean vector space adds further constraints on the color difference formula. When the dimension of the codomain is not specified, these constraints are difficult to characterize, but in the case where the codomain is of dimension three (i.e., of the same dimension as that of the original color space), a differential geometric characterization is available (cf. Reference 13, Thm. 7.10): The original color space endowed with the color difference formula as metric is isometrically embeddable into the Euclidean vector space \mathbb{R}^3 if and only if there exists a smooth structure^{13(Appendix A)} and a Riemannian metric^{13(Ch. 2)} on the color space turning the color space into a Riemannian manifold^{13(Ch. 2)} such that

- a. the given color difference formula is exactly the Riemannian distance function (i.e., the shortest path metric, cf. Reference 13, Thm. 2.55) of the Riemannian manifold and
- b. the Riemannian manifold has zero curvature^{13(Ch. 7)}.

Concerning condition (a), the ansatz of modeling the space of perceived colors as a Riemannian manifold is at least over a century old¹⁴ and many of today's color difference formulas can be seen to at least asymptotically (i.e., as color difference tends towards zero) agree with a Riemannian distance function. Concerning condition (b), for the CIELAB-based color difference formulas, CMC, CIE94, and DE2000, Urban et al.^{12(sec. 1.A)} analyzed the curvature of the ab-plane of CIELAB and showed that the corresponding Riemannian metrics exhibit nonzero Gaussian curvature, which poses an obstruction to the existence of an isometric embedding of CIELAB into \mathbb{R}^3 with respect to these color difference formulas.

In view of the above limitations while attempting to embed a color difference system isometrically into a Euclidean space, efforts have been made to identify nearly-isometric embeddings. That is, given a color difference formula as reference distance law, an injective transformation from the original color space into a Euclidean space is being sought, such that the deviation of the resulting Euclidean distances from their respective reference color distances is minimized with respect to a certain disagreement measure. Approaches to this type of color space embedding problem include, but are not limited to: analytical methods, for example, References 15,16, and numerical methods, for example, References 12,17. Among these, the analytical approaches basically aim at working out the antiderivative of the reference color difference formula, which is only feasible for rather simple color distance laws, whereas the numerical methods mainly rely on coupled local optimizations on meshes complemented by interpolation techniques.

This work continues the previous efforts on identifying mathematical approximations for embedding existing non-Euclidean color difference systems isometrically into Euclidean ones, referred to as color space Euclidization in the sequel. While the derivation of the color difference formula ΔE itself is based on experimental visual data (i.e., perceived color differences ΔV), the corresponding color space Euclidization constitutes a theoretical approach that is detached from actual visual experiments. Performance-wise, the perceptual non-uniformity of the Euclidized color difference system (i.e., deviation of $\Delta E_{\text{Euclidean}}$ from ΔV) is upper-bounded by the perceptual non-uniformity of the reference color difference formula (i.e., deviation of ΔE from ΔV) plus the isometric disagreement arising from the color space embedding (i.e.,

deviation of $\Delta E_{\text{Euclidean}}$ from ΔE). Therefore, if the isometric disagreement in the Euclidization step could be minimized, the resulting new color space would largely preserve the perceptual uniformity of the underlying color difference formula while benefiting from the inherent advantages of a vector space representation, which constitutes the general motivation of color space Euclidization. From a practical point of view, functioning as a perceptually uniform color space, the Euclidized color difference system would offer more possibilities than the original one endowed with non-Euclidean color difference formula in a number of applications. For instance, the vector space representation of perceptual uniformity would allow for linear interpolation between color coordinates within the new space while preserving the perceived color difference ratios, for example, the midpoint between two points within the new vector space could be used for representing the perceived average color of the two corresponding color samples. In the context of image processing, in contrast to the color difference formula merely serving the purpose of a metric, the invertible transformation associated with the color space Euclidization would allow for encoding color information into a Cartesian coordinate system to work with and decoding back into the original color coordinate system. Typical instances benefiting from this are represented by image quantization and lossy image compression, where transforming the initial color space (e.g., sRGB or CIELAB) into a linear working domain that is arranged in a perceptually uniform manner would allow for determining threshold values for just-noticeable or -tolerable color distance through equal-sized l^2 -spheres, based on which an efficient and perception-based information reduction could be implemented (cf. e.g., Reference 18, sec. 1 for a more detailed discussion).

In recent years, with the rising availability of computational power, machine learning techniques have come into use in a variety of research fields and real world applications such as image processing, biomedicine, and virtual reality. The main aim of this work is to explore the capability of such techniques in tackling the above color space Euclidization problem. For this, a machine learning approach making use of artificial neural networks is proposed herein. In general, neural networks are function approximators that typically are composed of intermediate mappings (layers), each combining an affine transformation with adjustable parameters (weights or kernels, and bias) and a fixed nonlinear transformation (activation function). Given an optimization criterion (loss function), the parameters of a neural network are adjusted to best fit all the available training examples through stochastic gradient descent (training), cf. for example, Reference 19. When applied to the present Euclidization task, the neural network under consideration represents a continuously

differentiable color space transformation. Concerning the optimization criterion employed during training, it is natural to define this in accordance with a pre-determined key performance metric. In the literature, the two most considered disagreement measures in the context of color space Euclidization are the relative isometric disagreement (cf. e.g., References 15,17) and the standardized residual sum of squares (STRESS) index (cf. e.g., References 12,20). While the former is a common relative error function, the latter is a statistical measure of absolute disagreement. Without attempting to judge which one of these disagreement indices is more appropriate for evaluating the efficacy of a nearly-isometric color space embedding, in this work, they are optimized separately through training and the resulting neural network color space transformations are evaluated for comparison. In this exposition of the method, the DE2000, CMC, and CIE94 color difference formulas are taken as example color distance laws for evaluation. Compared to previous contributions, the proposed machine learning approach applies to a broader range of settings and proves to be more powerful when evaluated with various disagreement metrics, regardless of the choice between the aforementioned training criteria.

In order to avoid confusion, it should be emphasized that the approach to color space Euclidization merely aims at achieving an alignment of the Euclidean metric on the embedded color space with the metric induced by the reference color distance law, which is independent of the efficacy of the underlying color difference formula itself in predicting actual visual differences.

The remainder of the article is organized as follows. In Section 2, a mathematical formulation of the problem of color space Euclidization is presented. Section 3 provides a general machine learning methodology for the considered Euclidization problem, including the architecture of the employed neural networks, the random sampling procedure for generating training examples, the loss functions associated with the aforementioned disagreement measures, and the training algorithm that is based on stochastic gradient descent and fortified with certain regularization measures. In Section 4, a series of example scenarios selected for training and evaluation are introduced. The layout of the neural network color space transformations and the training configurations employed for those example scenarios are specified in Section 5. The evaluation results are presented in Section 6. Section 7 includes the following discussions: the computational effort required during the inference stage of the proposed neural network approach is discussed in Section 7.1; the perceptual uniformity of the Euclidized DE2000 color difference system is evaluated on experimental visual data in Section 7.2; as an outlook, potential challenges that may be encountered when carrying over the present approach to the issue of color space

modeling with experimental visual data are addressed in Section 7.3; for expanding perspective, potential intersections of the present work with a selection of machine learning approaches to perception-based image processing issues are discussed in Section 7.4. Section 8 concludes the article.

Throughout this work, $\|\cdot\|$ refers to the Euclidean norm and bold symbols refer to multidimensional vectors or tensors.

2 | PROBLEM OF COLOR SPACE EUCLIDIZATION

Let $\Delta E: \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}_+$ be the reference color difference formula. Let $\text{err}: (0, \infty) \times (0, \infty) \rightarrow \mathbb{R}_+$, $(r, r') \mapsto \text{err}(r, r')$ denote the reference error function that measures the discrepancy between two metrics. For a color space transformation $f: \mathbb{R}^3 \rightarrow \mathbb{R}^d$ with $d \geq 3$, define the isometric disagreement function, $\text{disg}_{\Delta E}^f: \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}_+$, associated with the reference error function $\text{err}(\cdot, \cdot)$ via

$$\text{disg}_{\Delta E}^f(\mathbf{x}, \mathbf{y}) := \text{err}(\|f(\mathbf{x}) - f(\mathbf{y})\|, \Delta E(\mathbf{x}, \mathbf{y})),$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ with $\mathbf{x} \neq \mathbf{y}$. Further, let $R \subseteq \mathbb{R}^3$ denote the reference region and $P^{\Delta \mathbf{x}}$ denote the probability distribution of the test color difference vector in the original color space. For predetermined ΔE , err , R , and $P^{\Delta \mathbf{x}}$, and given the dimension of the codomain, $d \geq 3$, the problem of nearly-isometric color space embedding according to the reference distance law ΔE consists in finding an injective transformation, $f_{\Delta E}: \mathbb{R}^3 \rightarrow \mathbb{R}^d$, such that

$$\int_R \mathbb{E}_{\Delta \mathbf{x} \sim P^{\Delta \mathbf{x}}} \left[\left(\text{disg}_{\Delta E}^f(\mathbf{x}, \mathbf{x} + \Delta \mathbf{x}) \right)^2 \right] d\mathbf{x} \stackrel{!}{=} \min, \quad (1)$$

where, $\mathbb{E}_{\Delta \mathbf{x} \sim P^{\Delta \mathbf{x}}}$ refers to the expectation with respect to the probability measure $P^{\Delta \mathbf{x}}$. Correspondingly, the inverse mapping of $f_{\Delta E}$ is to be approximated by a transformation, $\tilde{f}_{\Delta E}: f_{\Delta E}(\mathbb{R}^3) \rightarrow \mathbb{R}^3$, such that

$$\int_R \left\| \left(\tilde{f}_{\Delta E} \circ f_{\Delta E} \right) (\mathbf{x}) - \mathbf{x} \right\|^2 d\mathbf{x} \stackrel{!}{=} \min. \quad (2)$$

In this work, either of the following two types of isometric disagreement function is considered at a time and incorporated as reference disagreement measure into the embedding problem:

- The first type is related to the *symmetric relative error*, $\text{err}: (r, r') \mapsto \max\{r, r'\} / \min\{r, r'\} - 1$. The corresponding *relative isometric disagreement function* reads as

$$\text{disg}_{\Delta E}^f(\mathbf{x}, \mathbf{y}) := \frac{\max\{\|f(\mathbf{x}) - f(\mathbf{y})\|, \Delta E(\mathbf{x}, \mathbf{y})\}}{\min\{\|f(\mathbf{x}) - f(\mathbf{y})\|, \Delta E(\mathbf{x}, \mathbf{y})\}} - 1, \quad (3)$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ with $\mathbf{x} \neq \mathbf{y}$, and $f: \mathbb{R}^3 \rightarrow \mathbb{R}^d$. In this case, problem (1) translates to minimizing the mean squared relative disagreement between the Euclidean distances in the embedded color space and the distances according to the reference color difference formula. The relative isometric disagreement is a common relative error function and has been incorporated as key performance measure into the objective function in various forms in previous contributions, cf. for example, References 15,17. Since the L^2 space lies midway between the L^1 and L^∞ spaces in the sense of interpolation theory (cf. Reference 21), the squared error minimization proposed herein can be considered as a compromise between the minimizations of mean and maximum relative isometric disagreement.

- The second type is related to the *absolute error*, $\text{err}: (r, r') \mapsto |r - r'|$. The corresponding *absolute isometric disagreement function* reads as

$$\text{disg}_{\Delta E}^f(\mathbf{x}, \mathbf{y}) := \left| \|f(\mathbf{x}) - f(\mathbf{y})\| - \Delta E(\mathbf{x}, \mathbf{y}) \right|, \quad (4)$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, and $f: \mathbb{R}^3 \rightarrow \mathbb{R}^d$. When taking this function as the key disagreement measure in (1) and approximating the double integral therein through Monte-Carlo simulation with a finite number of random samples (law of large numbers), the respective optimization problem translates to minimizing the standardized residual sum of squares (STRESS) index²⁰ while excluding any scale ambiguity inherent in that index. More details on this equivalence are provided in Section 3.4.2. In colorimetry, the STRESS index is considered a statistically more reliable alternative to the PF/3 index²² for evaluating the similarity between two color difference datasets (cf. Reference 20). Apart from the correspondence with the STRESS index, the mean squared absolute error is a common indicator for average discrepancy and has been employed as optimization criterion in previous contributions to the issue of color space Euclidization, cf. References 12,18.

Depending on the nature of the underlying color difference formula, the choice between the above reference disagreement measures as optimization criterion may have a notable impact on the resulting color space transformation, which will be elucidated in the evaluation section.

The introduction of the probability measure $P^{\Delta \mathbf{x}}$ in (1) mainly aims at a unified description of the Euclidization problem in terms of the reference range and weights of color distances. Typically, the color difference vector $\Delta \mathbf{x}$ in (1) is decomposed into its length r and direction \mathbf{v} , that is, $\Delta \mathbf{x} = r \cdot \mathbf{v}$, with r and \mathbf{v} independently distributed according to their respective probability measures P^r and $P^{\mathbf{v}}$. For instance, the direction component \mathbf{v} is usually assumed to follow the uniform distribution on the unit sphere $S^2 \subseteq \mathbb{R}^3$; the absolute color difference r may be uniformly distributed on an interval $(r_{\min}, r_{\max}]$ as in Reference 12, or normally distributed as in Reference 17 where the probability density function of P^r was used as weighting factor in the objective function.

3 | GENERAL METHODOLOGY

In this section, a general machine learning methodology is proposed for the color space Euclidization problem. The basic idea herein is to exploit the excellence of artificial neural networks in the task of function approximation and training these properly so as to obtain the desired color space transformations.

Being intended for readers who are not yet familiar with machine learning techniques, the general approach is presented in a self-contained manner, with most of the recurring basic notions and terminology regarding artificial neural networks and training principles introduced in a general fashion in Sections 3.1 and 3.2. Readers who are familiar with machine learning techniques may skip most parts of Sections 3.1 and 3.2, except for the thoughts behind the somewhat unconventional use of residual connections in-between fully connected layers in the present application (cf. second-last paragraph of Section 3.1). Furthermore, in order to enable the machine learning expert to quickly skim through Sections 3.3–3.5, some distinctive characteristics of the proposed approach tailored to the present Euclidization issue are highlighted as follows:

- Training relies on unsupervised learning through random sampling, which is based on Monte-Carlo simulation following the law of large numbers (cf. Section 3.3); in particular, no external labels are required and overfitting may not occur.
- In accordance with the distinction of cases in terms of the reference isometric disagreement measure (recall (3) and (4)), two different loss functions are employed for the corresponding optimization problems (cf. Section 3.4). In particular, for minimization with respect to the relative isometric disagreement, a non-obvious choice of loss function is made, which acts as

an infinitely differentiable proxy for the original non-differentiable (symmetric) relative disagreement function (cf. discussion in Section 3.4.1); for minimization with respect to the absolute isometric disagreement, the inherent relationship between the minimization of mean squared absolute disagreement and that of STRESS is elaborated (cf. discussion in Section 3.4.2).

- A composed training procedure is proposed where the two central optimization problems, (1) and (2), are considered jointly and tackled in alternation through certain regularization measures, which aids in preventing potential invertibility issues and speeding up training (cf. Section 3.5).

3.1 | Feedforward neural network with residual connections

In the present application, fully connected neural networks with feedforward architecture augmented by residual connections are employed. Subsequently, a precise description of the related transformations is provided.

Let f_{NN} be the output function of a neural network and let $\mathbf{u}^{\text{in}} \in \mathbb{R}^I$, $\mathbf{u}^{\text{out}} \in \mathbb{R}^J$, and $\boldsymbol{\xi} \in \mathbb{R}^P$ denote the input and output vectors, and the vector of all parameters of the neural network, respectively. For a basic feedforward fully connected neural network, the output function is of the form

$$f_{\text{NN}} : \mathbb{R}^I \times \mathbb{R}^P \rightarrow \mathbb{R}^J$$

$$(\mathbf{u}^{\text{in}}, \boldsymbol{\xi}) \mapsto \mathbf{u}^{\text{out}}, \quad (5)$$

with

$$f_{\text{NN}} = f^{(M-1)} \circ \dots \circ f^{(0)}, \quad (6)$$

where each intermediate layer, $f^{(m)} : \mathbb{R}^{I^{(m)}} \rightarrow \mathbb{R}^{J^{(m)}}$ for $m = 0, \dots, M-1$, is a transformation parametrized through

$$\mathbf{u}^{(m)} \mapsto \mathbf{u}^{(m+1)} = \left\{ f_{\text{actv}}^{(m)} \left(\sum_{i=0}^{I^{(m)}-1} u_i^{(m)} w_{ij}^{(m)} + b_j^{(m)} \right) \right\}_{j=0}^{J^{(m)}-1}, \quad (7)$$

with $\mathbf{w}^{(m)} \in \mathbb{R}^{I^{(m)} \times J^{(m)}}$ and $\mathbf{b}^{(m)} \in \mathbb{R}^{J^{(m)}}$ referred to as *weight matrix* and *bias vector*, respectively, and $f_{\text{actv}}^{(m)} : \mathbb{R} \rightarrow \mathbb{R}$ representing the *activation function* of the layer, which is typically defined as a nonlinear almost-everywhere differentiable function operating component-wise.

In particular, it holds that $\mathbf{u}^{(0)} = \mathbf{u}^{\text{in}}$, $\mathbf{u}^{(M)} = \mathbf{u}^{\text{out}}$, $\xi = \left\{ \left(\mathbf{w}^{(m)}, \mathbf{b}^{(m)} \right) \right\}_{m=0}^{M-1}$, and $J^{(m)} = I^{(m+1)}$ for all $m = 0, \dots, M - 2$. In (7), the output dimension, $J^{(m)}$, is also called the *size* or *number of channels* of the layer; the input dimension of the first layer, I , and the output dimension of the last layer, J , are referred to as the *number of input channels* and *number of output channels* of the neural network, respectively. A layer whose output is not the final output of the entire neural network, that is, $f^{(m)}$ for $m < M - 1$ in (6), is called a *hidden layer*. The total number of layers, M , is referred to as the *depth* of the neural network.

For the present Euclidization task, the solutions to the optimization problems (1) and (2) are each approximated by a feedforward neural network where the numbers of input and output channels of the neural network are set to $(I := 3, J := d)$ for the embedding problem (1) and vice versa, $(I := d, J := 3)$ for the inverse problem (2). Throughout the neural network transformation (except for at the output layer), the hyperbolic tangent function, \tanh , is chosen as the activation function, which is defined as

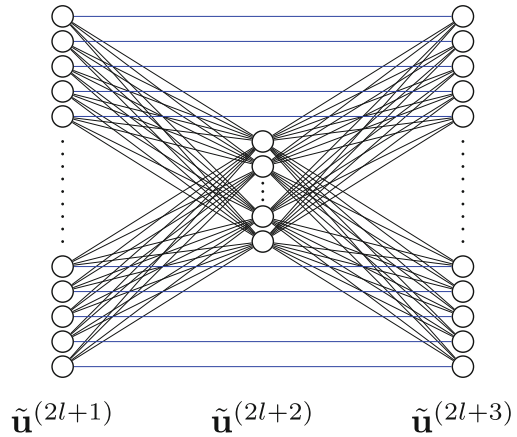
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \text{ for } x \in \mathbb{R}.$$

Compared to other common choices such as the rectified linear unit (ReLU), the hyperbolic tangent is an analytic function, which implies the infinitely differentiable nature of the resulting color space transformation, as opposed to the piecewise-smooth transformations devised in previous contributions.^{12,17}

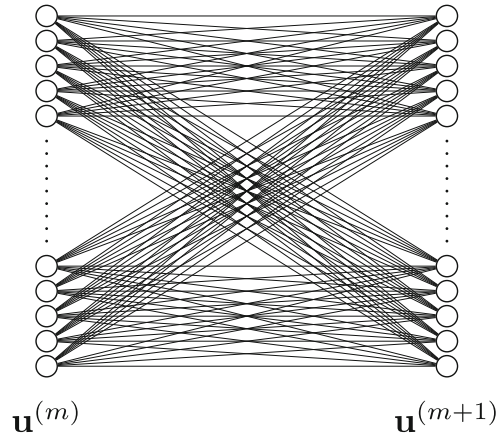
On top of the basic fully connected layers, the hidden layers employed in the present application are endowed with so-called *residual connections*. Keeping the above notations for the parameters and layer indices and assuming that M is an even number and the sizes of channels satisfy $I^{(2l+1)} = J^{(2l+2)}$ for all $l = 0, \dots, M/2 - 2$, each group of two adjacent layers is considered as a building block, called *residual layer*, and the input vector to each block, denoted by $\tilde{\mathbf{u}}^{(2l+1)}$, is additionally propagated through shortcut connections within the block, as illustrated in Figure 1A. The transformation of the residual layer indexed by $[2l + 1, 2l + 2]$ for each l then reads as $\tilde{\mathbf{u}}^{(2l+1)} \mapsto \tilde{\mathbf{u}}^{(2l+3)}$ with

$$\tilde{\mathbf{u}}^{(2l+3)} := f_{\text{actv}}^{(2l+2)} \left(\left(\tilde{\mathbf{u}}^{(2l+2)} \cdot \mathbf{w}^{(2l+2)} + \mathbf{b}^{(2l+2)} \right) + \tilde{\mathbf{u}}^{(2l+1)} \right),$$

where, $\tilde{\mathbf{u}}^{(2l+1)} \mapsto \tilde{\mathbf{u}}^{(2l+2)}$ follows the same transformation as in (7).



(A) Residual building block with small inner layer



(B) Densely connected layers

FIGURE 1 Comparison: residual versus densely connected layers.

Originally, residual connections were introduced for the purpose of improving gradient propagation within a deep convolutional neural network, cf. Reference 23. In the present application, on the other hand, employing residual layers mainly aims at a relatively parsimonious parametrization for the sake of saving computational effort. More specifically, as illustrated through the transition from (B) to (A) in Figure 1, when inserting a small mid-layer between two adjacent fully connected layers of equal size, the total number of weight parameters is reduced, as long as the in-between layer is smaller than half the size of the outer layers. However, because of the dimensionality reduction in the mid-layer, employing such a layout without modification entails the so-called bottleneck effect in general and may lead to a decline in performance; as a countermeasure, introducing skip connections between the outer layers within such an architecture aids in mitigating the bottleneck effect while maintaining the sleekness of the original reduced-size

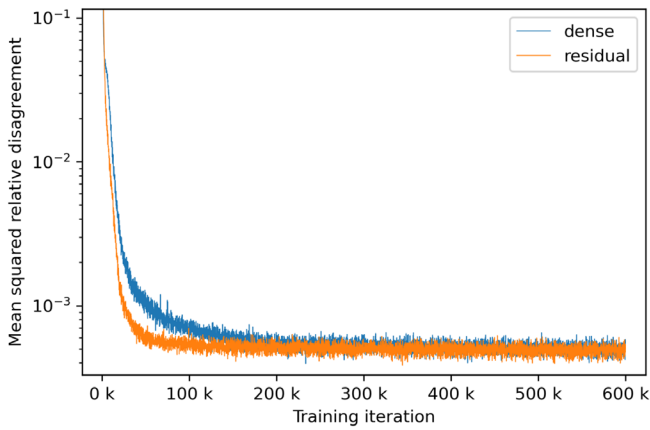


FIGURE 2 Course of mean squared relative isometric disagreement while training embedding neural networks according to the DE2000 color difference formula, where the hidden layers consist of either two densely connected layers of size 45:45 or a residual building block of size 45:15:45.

layout, which in particular yields a residual building block as introduced above (cf. Reference 23 for a similar argument). Moreover, from a heuristic point of view, the present Euclidization problem can be thought of as looking for a neural network color space transformation that comprises some linear distortions (e.g., linear embedding or shear) and a (higher-dimensional) nonlinearly distorted identity mapping, which, respectively, can be expressed in terms of the linear parts of the input and output layers, and the operations within a residual building block. In this sense, the aforementioned potential bottle-neck issue mainly affects the component of nonlinear distortions within the entire color space transformation, which is not expected to cause a significant increase in the overall isometric disagreement. In fact, as depicted in Figure 2, experimentation in this work reveals that, despite the concerns over its inherent sparsity, the proposed residual architecture even exhibits slightly better performance and faster convergence compared to its densely connected counterpart.

The neural network layout employed in the example scenarios for evaluation is determined in consideration of performance, computational efficiency, and training speed, which will be presented in Section 5.1 and discussed in Section 7.1.

3.2 | Basic principles of stochastic gradient descent

The algorithm proposed for training the neural network color space transformations is built on *stochastic gradient descent* (SGD). Basically, training with SGD requires an

almost-everywhere differentiable neural network output function with respect to all parameters (which is fulfilled in the architecture introduced in Section 3.1), a set of training examples, and an almost-everywhere differentiable loss function.

The set of *training examples* consists of pairs of input vectors to the neural network and target values towards which the neural network is to be trained. The total number of available training examples may be finite, as is the case with supervised learning where a fixed training dataset is randomly partitioned into *mini-batches* during each training *epoch*, or infinite, as is the case with the present approach where a mini-batch of training examples is generated through random sampling for each parameter update (unsupervised learning). A detailed description of the data generation procedure will be presented in Section 3.3.

A *loss function* measures the distance of the neural network output from the training target and serves as an indicator for the current approximation error. In the context of color space Euclidization, the shape of the loss function depends on the choice of the reference disagreement measure (recall (3) and (4)) as well as possible regularization measures. This will be further specified in Sections 3.4 and 3.5.

In SGD, the loss function is evaluated mini-batch-wise and minimized iteratively by moving the parameter vector of the neural network in the direction of the negative gradient. Let $f_{\text{NN}}(\cdot, \xi)$ be the neural network output function with parameter vector ξ (recall (5)) and let f_{loss} denote the loss function. In the update step with index t , given the corresponding mini-batch of $N(t)$ -many training examples denoted by $\{(\mathbf{u}^{\text{in},t,n}, \mathbf{u}^{\text{target},t,n})\}_{n=0}^{N(t)-1}$, the parameter vector ξ is adjusted through

$$\xi \leftarrow \xi - \frac{\gamma}{N(t)} \sum_{n=0}^{N(t)-1} \nabla_{\xi} f_{\text{loss}}(f_{\text{NN}}(\mathbf{u}^{\text{in},t,n}, \xi), \mathbf{u}^{\text{target},t,n}),$$

where, $\gamma > 0$ is a tunable hyperparameter, referred to as *learning rate*, and the gradient can be computed by means of *backpropagation* which is an instance of reverse mode automatic differentiation based on the chain rule, cf. for example, Reference 19, Ch. 4. Throughout all experiments in this work, the Adam algorithm is employed, which is a refined version of SGD, incorporating an adaptive moment estimator into the update rule (cf. Reference 24 for more details).

Following the above general principles of training with SGD, the upcoming subsections are devoted to the individual components of the training procedure devised for the present Euclidization problem.

3.3 | Generation of training examples

In the proposed machine learning approach, training examples for both the color space embedding problem and the corresponding inverse approximation are generated through random sampling on the original color space.

For the color space embedding problem, random samples are generated with the aim of approximating the double integral in (1) through Monte-Carlo simulation (law of large numbers), which is performed as follows:

1. For each mini-batch indexed by ι , draw independently $N(\iota)$ -many color centers, $\{\mathbf{x}^{i,n}\}_{n=0}^{N(\iota)-1}$, according to the uniform probability distribution on the reference region $R \subseteq \mathbb{R}^3$.
2. For each color center $\mathbf{x}^{i,n} \in R$, draw independently $K(\iota)$ -many color difference vectors, $\{\Delta \mathbf{x}^{i,n,k}\}_{k=0}^{K(\iota)-1}$, according to the probability distribution of the test color difference vector¹ $P^{\Delta \mathbf{x}}$.
3. Collect the above samples for all indices n and k into a tensor of training inputs for mini-batch ι , $\left\{ \left\{ \mathbf{u}^{i,n,k} \right\}_{k=0}^{K(\iota)} \right\}_{n=0}^{N(\iota)-1}$, through

$$\begin{aligned} \mathbf{u}^{i,n,k} &:= \mathbf{x}^{i,n} + \Delta \mathbf{x}^{i,n,k} \text{ for } k < K(\iota) \\ \mathbf{u}^{i,n,K(\iota)} &:= \mathbf{x}^{i,n} \end{aligned} \quad (8)$$

When training a neural network color space embedding with respect to a given color difference formula ΔE , in each update step ι , the samples associated with each of the selected color centers, $\{\mathbf{u}^{i,n,k}\}_{k=0}^{K(\iota)}$ in (8) for an index n , are processed through the embedding neural network, from which the Euclidean distances of the transformed samples from the transformed color center are computed. The corresponding training loss is then determined by the disagreement of these Euclidean distances with their respective reference color distances, $\{\Delta E(\mathbf{u}^{i,n,k}, \mathbf{u}^{i,n,K(\iota)})\}_{k=0}^{K(\iota)-1}$, where the latter serve as the training targets associated with the color center $\mathbf{x}^{i,n} = \mathbf{u}^{i,n,K(\iota)}$.

Similarly, for the inverse approximation, training examples are generated through random sampling following Step 1 of the above procedure, so as to approximate the integral in (2) through Monte-Carlo simulation. While being processed through the composition of the trained embedding neural network and the current inverse neural network, the samples drawn from the original color space serve as the inverse training targets and the inverse training loss is determined by the total transformation error of the joint neural network compared to the identity mapping.

3.4 | Isometric disagreement functions for training loss

Henceforth, for a fixed reference color difference formula ΔE , let $f_{\Delta E}(\cdot, \xi)$ and $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$ denote the neural network output functions with parameter vectors ξ and $\tilde{\xi}$ employed for solving the embedding problem (1) and the corresponding inverse approximation (2), respectively.

As introduced in Section 3.2, training a neural network relies on the SGD algorithm, which requires primarily the specification of an almost-everywhere differentiable loss function. In general, it is reasonable to define the loss function in accordance with the key performance metric being included in the optimization criterion. Based on these considerations, when training the embedding neural network $f_{\Delta E}(\cdot, \xi)$ for solving problem (1), either of the following isometric disagreement functions is considered at a time and employed for computing the training loss:

- When minimizing in (1) the mean squared relative isometric disagreement associated with (3), the *logarithmic error function*, $\text{err}: (r, r') \mapsto |\ln(r/r')|$, serves as a proxy for the original symmetric relative error function during training. The corresponding isometric disagreement function is defined as

$$\text{disg}_{\Delta E}(\mathbf{x}, \mathbf{y}, \xi) := \left| \ln \frac{\|f_{\Delta E}(\mathbf{x}, \xi) - f_{\Delta E}(\mathbf{y}, \xi)\|}{\Delta E(\mathbf{x}, \mathbf{y})} \right|, \quad (9)$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ with $\mathbf{x} \neq \mathbf{y}$.

- When minimizing in (1) the mean squared absolute isometric disagreement associated with (4), the isometric disagreement function employed during training reads as

$$\begin{aligned} \text{disg}_{\Delta E}(\mathbf{x}, \mathbf{y}, \xi) &:= \left| \|f_{\Delta E}(\mathbf{x}, \xi) - f_{\Delta E}(\mathbf{y}, \xi)\| - \Delta E(\mathbf{x}, \mathbf{y}) \right|, \end{aligned} \quad (10)$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$.

When training the inverse neural network transformation $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$ for solving problem (2), the inverse training loss is determined by means of the total transformation error function, defined as

$$\widetilde{\text{disg}}_{\Delta E}(\mathbf{x}, \xi, \tilde{\xi}) := \|\tilde{f}_{\Delta E}(f_{\Delta E}(\mathbf{x}, \xi), \tilde{\xi}) - \mathbf{x}\|, \quad (11)$$

for all $\mathbf{x} \in \mathbb{R}^3$.

In each parameter update step during training, the respective disagreement function introduced above is evaluated on the current mini-batch of random color

coordinates generated according to Section 3.3, from which the mean squared value is determined and minimized as training loss through gradient descent. The rigorous parameter update rules including possible regularization measures will be presented shortly (in Section 3.5). Prior to this, the following explanatory comments are provided for justifying the choice of the isometric disagreement functions (9) and (10) in their respective optimization contexts:

3.4.1 | Notes on minimization of mean squared relative disagreement

In (9), the original symmetric relative error function, $(r, r') \mapsto \max\{r, r'\} / \min\{r, r'\} - 1$, is replaced by the logarithmic error function, $(r, r') \mapsto |\ln(r/r')|$, due to the following consideration: The symmetry of the symmetric relative error in its arguments relies on the distinction of cases inherent in the min and max expressions, whereas the logarithmic error exhibits the same type of symmetry while having an infinitely differentiable square; employing the latter for gradient descent aids in avoiding potential instabilities during training. Note that both of these error functions are directly determined by the quotient of their arguments r/r' (as opposed to the absolute error function) and minimized at $r = r'$, and their values are close to one another near their joint minimum, as illustrated in Figure 3.

3.4.2 | Notes on minimization of mean squared absolute disagreement

When evaluated on a finite number of samples, minimizing the mean squared absolute disagreement corresponds to minimizing the STRESS index. To

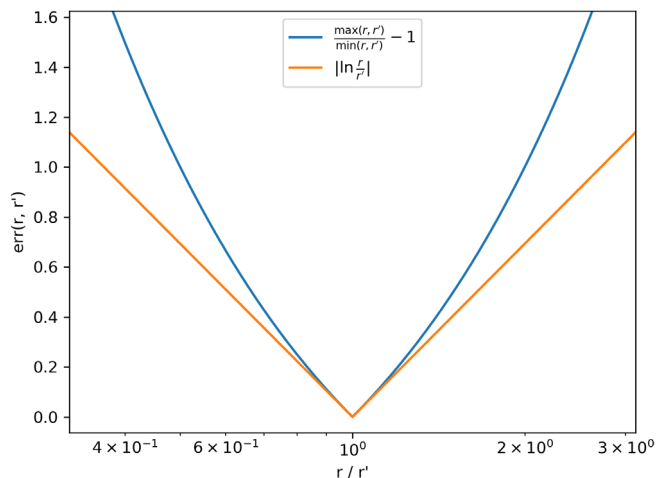


FIGURE 3 Illustration of relative error functions.

demonstrate this correspondence, let $\{\widehat{\Delta E}_i\}_{i \in I}$ and $\{\Delta E_i\}_{i \in I}$ denote the estimated and reference color differences evaluated on a dataset indexed with $i \in I$, respectively. According to one of the equivalent definitions given in Reference 20, eq. (9), the STRESS index between $\{\widehat{\Delta E}_i\}_{i \in I}$ and $\{\Delta E_i\}_{i \in I}$ reads as

$$\text{STRESS}\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right) := \left(\frac{\sum_{i \in I} \left(F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right) \widehat{\Delta E}_i - \Delta E_i \right)^2}{\sum_{i \in I} \Delta E_i^2} \right)^{1/2}, \quad (12)$$

where the scale factor $F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)$ is defined as

$$F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right) := \frac{\sum_{i \in I} \widehat{\Delta E}_i \Delta E_i}{\sum_{i \in I} \Delta E_i^2}.$$

It is a straightforward exercise to show that the definition of $F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)$ is such that the associated STRESS index becomes minimal. This is done to remove any dependency of $\text{STRESS}\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)$ on the global scale of $\left\{\widehat{\Delta E}_i\right\}_{i \in I}$. If $F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)$ was set to 1 instead, the denominator in Equation (12) would reduce to the sum of squared absolute errors; in particular, $\text{STRESS}\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)^2$ is always bounded from above by the mean squared absolute error scaled by the normalizing factor $\#I / \sum_{i \in I} \Delta E_i^2$ which does not depend on $\left\{\widehat{\Delta E}_i\right\}_{i \in I}$. In fact, minimizing the mean squared absolute error is equivalent to minimizing the STRESS index and scaling $\left\{\widehat{\Delta E}_i\right\}_{i \in I}$ such that $F_2\left(\left\{\widehat{\Delta E}_i\right\}_{i \in I}, \left\{\Delta E_i\right\}_{i \in I}\right)$ becomes 1, that is, that the global scale of $\left\{\widehat{\Delta E}_i\right\}_{i \in I}$ matches that of $\left\{\Delta E_i\right\}_{i \in I}$.

When employed as a loss function, the mean squared absolute disagreement based on (10) is preferred over the STRESS index, as the scale ambiguity introduced by the normalizing scale factor F_2 would lead to a subsequent target ambiguity in the output of the neural network transformation and thereby entails the potential to destabilize training. The additional constraint on the global scale of the color difference estimates does not introduce a significant extra burden to the training of the

embedding neural network, as it can trivially be satisfied by rescaling the parameters of the output layer.

3.5 | Training procedure

In the subsequent description of the algorithm, all notations are carried over from Sections 3.3 and 3.4.

Basically, the joint neural network, $\tilde{f}_{\Delta E}(f_{\Delta E}(\cdot, \xi), \tilde{\xi})$, can be considered as an autoencoder² where color coordinates in the original color space are encoded into their counterparts in the embedded space through the forward path, $f_{\Delta E}(\cdot, \xi)$, and then to be decoded back into themselves through the inverse path, $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$. Due to the consideration of potential invertibility issues, the minimization of the inverse training loss is incorporated as regularization measures into the training routine for the forward path. This can be expressed in terms of two alternating subroutines that are characterized by their respective parameter update rules as follows:

- R1.** Adjust the parameters of the forward path, $f_{\Delta E}(\cdot, \xi)$, by minimizing the forward loss. That is, for the current mini-batch indexed by ι , update the parameter vector ξ via

$$\xi \leftarrow \xi - \gamma \cdot \nabla_{\xi} \text{loss}^{\iota}(\xi),$$

with $\gamma > 0$ and

$$\text{loss}^{\iota}(\xi) := \text{mean}_{\substack{n=0, \dots, N(\iota)-1 \\ k=0, \dots, K(\iota)-1}} \text{disg}_{\Delta E}(\mathbf{u}^{\iota, n, k}, \mathbf{u}^{\iota, n, K(\iota)}, \xi)^2,$$

where, $\text{disg}_{\Delta E}$ refers to the isometric disagreement function defined in (9) or (10), and $\{\{\mathbf{u}^{\iota, n, k}\}_{k=0}^{K(\iota)}\}_{n=0}^{N(\iota)-1}$ are the randomly generated training samples for mini-batch ι as specified in (8) in Section 3.3.

- R2.** Adjust the parameters of the inverse path, $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$, while, if applicable, regularizing the forward path, $f_{\Delta E}(\cdot, \xi)$. That is, for the current mini-batch ι , minimize the inverse loss through

$$\begin{pmatrix} \tilde{\xi} \\ \xi \end{pmatrix} \leftarrow \begin{pmatrix} \tilde{\xi} \\ \xi \end{pmatrix} - \begin{pmatrix} \tilde{\gamma} \cdot \nabla_{\tilde{\xi}} \widetilde{\text{loss}}^{\iota}(\tilde{\xi}, \xi) \\ \beta \cdot \nabla_{\xi} \widetilde{\text{loss}}^{\iota}(\tilde{\xi}, \xi) \end{pmatrix}, \quad (13)$$

with $\tilde{\gamma} > 0, \beta \geq 0$, and

$$\widetilde{\text{loss}}^{\iota}(\tilde{\xi}, \xi) := \text{mean}_{n=0, \dots, N(\iota)-1} \widetilde{\text{disg}}_{\Delta E}(\mathbf{x}^{\iota, n}, \xi, \tilde{\xi})^2,$$

where, $\widetilde{\text{disg}}_{\Delta E}$ refers to the total transformation error function defined in (11) and $\{\mathbf{x}^{\iota, n}\}_{n=0}^{N(\iota)-1} \subseteq R$ are the training samples of mini-batch ι generated for the inverse approximation (recall last paragraph of Section 3.3).

In (13), the factor β acts as a tuning parameter that balances the proportion between the actual training loss and the regularization term.

Starting off by training the embedding neural network, $f_{\Delta E}(\cdot, \xi)$, Subroutines R1 and R2 are run block-wise and alternately, each for a pre-determined number of iterations within the current rotation. The update of parameters ξ is terminated if the performance of $f_{\Delta E}(\cdot, \xi)$ is considered satisfactory (e.g., compared to baseline methods) or no longer exhibits significant improvement. Subsequently, if necessary, the inverse transformation $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$ is further trained by only iterating Subroutine R2 with $\beta := 0$ in (13), until the total transformation accuracy of $\tilde{f}_{\Delta E}(f_{\Delta E}(\cdot, \xi), \tilde{\xi})$ reaches a predetermined reference value. The entire training procedure is summarized in a concise form in Algorithm 1. In the pseudo code, the integers $I, \#R1(i)$, and $\#R2(i)$ are hyperparameters that are to be tuned for each individual setting. Note that in the case where $\beta := 0$ in (13) and the “while loop” in Algorithm 1 is only run for a single iteration with $I := 1$, the corresponding routine translates to training the embedding and inverse neural network transformations separately in succession without any regularization measures.

The introduction of the above regularization measures primarily aims to prevent the embedding neural network from being driven towards a noninjective transformation. The key idea behind this consists in taking the performance of the potential inverse transformation, $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$, as an indicator for the regularity (in particular the injectivity) of the embedding transformation, $f_{\Delta E}(\cdot, \xi)$. In fact, when looking for a nearly-isometric color space embedding by iterating Subroutine R1 alone, the resulting neural network $f_{\Delta E}(\cdot, \xi)$ is trained to emulate the potentially rather complex characteristics of the underlying reference color difference formula ΔE , which, with rising number of iterations, increases the difficulty of training the corresponding inverse transformation, $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$. In the worst-case scenario, training in this manner may lead to local irregularities in the form of large local approximation errors or even local

ALGORITHM 1 Training algorithm.

```

set number of rotations per training block  $I$ 
 $i_0 \leftarrow 0$ 
while parameters of  $f_{\Delta E}(\cdot, \xi)$  or  $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$  need adjustment
do
  for  $i \in \{i_0, \dots, i_0 + I - 1\}$  do
    set  $\#R1(i)$ ,  $\#R2(i)$  depending on performance
    for  $j \in \{0, \dots, \#R1(i) - 1\}$  do
      generate training samples for mini-batch  $(i, j)$ ,
       $\{\{\mathbf{u}^{(i,j),n,k}\}_{k=0}^{K(i,j)}\}_{n=0}^{N(i,j)-1}$ ,
      according to Steps 1), 2), 3) in Section 3.3
      update parameters  $\xi$  of neural network  $f_{\Delta E}(\cdot, \xi)$ 
      according to Subroutine R1
    end for
    for  $j \in \{\#R1(i), \dots, \#R1(i) + \#R2(i) - 1\}$  do
      generate training samples for mini-batch  $(i, j)$ ,
       $\{\mathbf{x}^{(i,j),n}\}_{n=0}^{N(i,j)-1}$ ,
      by randomly sampling from  $R \subseteq \mathbb{R}^3$ 
      update parameters  $\tilde{\xi}$  of  $\tilde{f}_{\Delta E}(\cdot, \tilde{\xi})$  and,
      if applicable, update parameters  $\xi$  of  $f_{\Delta E}(\cdot, \xi)$ 
      according to Subroutine R2
    end for
  end for
   $i_0 \leftarrow i_0 + I$ 
end while

```

[Correction added on 11 October 2023, after first Online publication: The formatting of Algorithm 1 has been corrected.]

noninjectivities. Such irregularities may occur, for instance, when it is possible to achieve a low average loss over the entire color space at the price of sacrificing the performance of the transformation $f_{\Delta E}(\cdot, \xi)$ within a few extremely small regions, cf. Appendix A for an example. As a countermeasure, through the alternation with Subroutine R2 and in particular the regularization term therein (governed by factor β), the joint transformation $\tilde{f}_{\Delta E}(f_{\Delta E}(\cdot, \xi), \tilde{\xi})$ considered as a whole is driven towards the identity mapping and thereby, the forward path $f_{\Delta E}(\cdot, \xi)$ is more likely to remain injective. It is worth noting that training the forward and inverse paths of the autoencoder $\tilde{f}_{\Delta E} \circ f_{\Delta E}$ alternately as proposed is reminiscent of the approach with generative adversarial nets (GANs)²⁵ in the sense that the two component neural networks, $f_{\Delta E}$ and $\tilde{f}_{\Delta E}$, indeed operate against one another during training. Figure 4 illustrates the interaction between these two neural networks during the alternations between Subroutines R1 and R2. Each time switching to Subroutine R1, the performance of

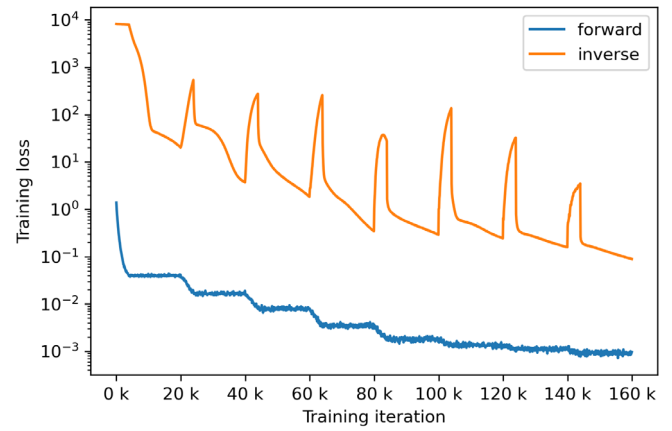
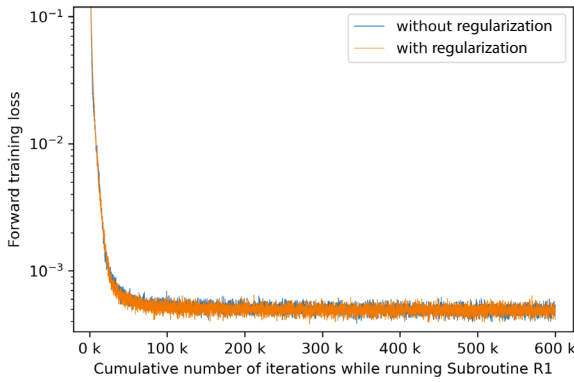


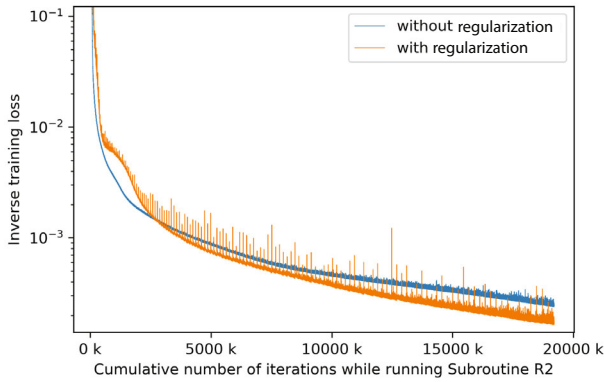
FIGURE 4 Course of forward and inverse losses while training an embedding neural network according to the DE2000 color difference formula; only the first 8 rotations of Algorithm 1 are depicted in the figure, where the forward-to-inverse ratio for regularization is set to $[\#R1(i) : \#R2(i)] := [4k : 16k]$ for $i < 8$.

the inverse path degrades with the growing complexity of the forward path; vice versa, each time switching to Subroutine R2, the minimization of the forward loss is temporarily halted and small adjustments of the forward path in favor of the inverse training are triggered instead.

In general, the occurrence of invertibility issues depends on the nature of the color difference formula and the composition of the objective function. Appendix A provides an instance of such where imposing the proposed regularization measures is essential to the injectivity of the embedding neural network transformation. In cases where the invertibility issue is not of concern, the regularization measures are optional; however, when configured properly, including them in the training routine may significantly speed up the inverse training (typically time consuming), without compromising the performance of the embedding transformation, as demonstrated in Figure 5. (Notice that in Figure 5A, the two courses of forward training loss hardly differ from one another, which indicates that the imposed regularization does not have a negative impact on the embedding neural network; on the other hand, in Figure 5B, the inverse training loss decreases faster through the regularization measures.) Ideally, the progress of the inverse training is spread over the course of the forward loss minimization such that the parameter updates of both the embedding and inverse transformations are terminated simultaneously. Similarly to GANs, the crucial point here is to figure out an appropriate proportion between the forward and inverse routines within each rotation (i.e., $[\#R1(i) : \#R2(i)]$) coupled with



(A) Forward training losses recorded while iterating Subroutine R1



(B) Inverse training losses recorded while iterating Subroutine R2; each spike in the orange curve indicates a switchover from Subroutine R1 to Subroutine R2 during regularization

FIGURE 5 Comparison: including versus omitting the proposed regularization measures while training the embedding and inverse color space transformations according to the DE2000 color difference formula.

$(\gamma, \tilde{\gamma}, \beta)$. This will be clarified for the selected example scenarios in Section 5.2.

4 | EXAMPLE SCENARIOS FOR TRAINING AND EVALUATION

In order to assess the efficacy of the proposed machine learning approach, the following instances of the Euclidization problem introduced in Section 2 are considered for training and evaluation:

As example reference color distance law, ΔE , the CIE-LAB-based color difference formulas, DE2000,¹⁰ CMC,⁶ and CIE94⁸ are considered. The default dimension of the embedding's codomain is $d := 3$. For comparative purposes and in accordance with Reference 12, the reference region is set to $R := [0, 100] \times [-128, 128]^2$ in the CIELAB coordinate space and the probability distribution of the test color difference vector, $P^{\Delta x}$, is defined as that of the

random vector $\Delta \mathbf{x} = r \cdot \mathbf{v}$ where the absolute color difference r follows the uniform distribution on $[r_{\min}, r_{\max}]$ (in CIELAB-units) and the direction vector \mathbf{v} follows the uniform distribution on the unit sphere $S^2 \subseteq \mathbb{R}^3$. Since the selected reference color difference formulas are all recommended for predicting small color differences and only nonzero color difference is admissible for evaluating the (symmetric) relative isometric disagreement (recall (3)), the interval $[r_{\min}, r_{\max}] := [\varepsilon, 5]$ with $\varepsilon := 10^{-3}$ is taken as the default range of the absolute color difference r . Scenarios associated with the above default settings are considered as primary cases, for which neural network color space transformations are trained with respect to both the relative and the absolute disagreement measures (each at a time). On top of the primary scenarios, for experimental purposes, embedding neural networks are also trained for color differences within $[r_{\min}, r_{\max}] := [5, 20]$ according to the DE2000 and CIE94 color difference formulas, despite the lack of a definitive statement on the efficacy of these formulas in predicting large color differences. Furthermore, as a dimensional analysis, embeddings with codomains of higher dimensions, $d = 4, 5, 6$, are explored for the DE2000 color difference formula, where only the mean squared relative isometric disagreement is minimized through training.

Concerning the training data generation for the example scenarios, in accordance with the above selected probability distribution of the test color difference vector, $P^{\Delta x}$, Step 2 in Section 3.3 is implemented through the following procedure:

- i. For mini-batch ι and for each of the color centers $\{\mathbf{x}^{i,n}\}_{i=0}^{N(\iota)-1} \subseteq R$, draw independently $K(\iota)$ -many directions, $\{\mathbf{v}^{i,n,k}\}_{k=0}^{K(\iota)-1}$, according to the uniform probability distribution on the unit sphere $S^2 \subseteq \mathbb{R}^3$. More specifically, for each coordinate $i = 0, 1, 2$, draw independently $K(\iota)$ -many samples, $\{\tilde{v}_i^{i,n,k}\}_{k=0}^{K(\iota)-1}$, according to the standard normal distribution $\mathcal{N}(0, 1)$; then, record $\mathbf{v}^{i,n,k} := (\tilde{v}_0^{i,n,k}, \tilde{v}_1^{i,n,k}, \tilde{v}_2^{i,n,k}) / \|(\tilde{v}_0^{i,n,k}, \tilde{v}_1^{i,n,k}, \tilde{v}_2^{i,n,k})\|$ for $k = 0, \dots, K(\iota) - 1$ (cf. Reference 26 for a derivation of this method).
- ii. For each direction $\mathbf{v}^{i,n,k} \in S^2$ around each color center $\mathbf{x}^{i,n}$, draw a random number, $r^{i,n,k}$, according to the uniform probability distribution on the interval $[r_{\min}, r_{\max}]$ and record $\Delta \mathbf{x}^{i,n,k} := r^{i,n,k} \cdot \mathbf{v}^{i,n,k}$.

The mini-batch sizes determined by $N(\iota)$ and $K(\iota)$ are hyperparameters for training and will be specified in Section 5.2.

Similarly to the training data generation, as in Reference 12, 4 million test samples (i.e., 2 million color pairs) are drawn randomly from the CIELAB color space for evaluation, which is performed through Step 1 in Section 3.3 complemented by the above procedure, with $N(\iota) := 2 \cdot 10^6$ and $K(\iota) := 1$ for $\iota := 0$ fixed. Among the generated test samples, the color centers are also used for evaluating the inverse approximation. In total, there are two test datasets, each containing 2 million color pairs generated as above, one for example scenarios associated with small color differences from the range $[r_{\min}, r_{\max}] := [10^{-3}, 5]$, one for those associated with large color differences from the range $[r_{\min}, r_{\max}] := [5, 20]$ (all in CIELAB-units). Note that as mentioned at the beginning of Section 3, the unsupervised approach to training through random sampling eliminates the danger of overfitting. This contrasts with most other machine learning problems where training data may inadvertently introduce bias in the resulting neural network. In the present case, however, the law of large numbers ensures that the objective functions in (1) and (2) will be approached arbitrarily closely as the number of random samples drawn for approximating the expectation and integrals therein tends to infinity. In particular, as random training examples keep being generated over the course of the entire training procedure, the training data do not introduce bias. For this reason, it is legitimate to indeed generate the test dataset according to the same procedure as the training data.

5 | NEURAL NETWORK LAYOUT AND TRAINING CONFIGURATION FOR EXAMPLE SCENARIOS

In this section, the layout of the neural network color space transformations and the hyperparameters of the training algorithm are specified for the example scenarios introduced in Section 4. In general, the optimal configuration may not be unique. The tuning work herein merely aims at finding one configuration per example that exhibits decent performance, so as to demonstrate the efficacy of the proposed approach. As a guideline, the neural network layout and the hyperparameter choice for all selected example scenarios are determined through minor adjustment of the configuration for the default setting associated with the most recognized and most complex color difference formula, DE2000. All neural networks and training routines are implemented using the PyTorch machine learning framework.²⁷

TABLE 1 Neural network layout for embedding ab-plane of CIELAB into \mathbb{R}^{d-1} according to DE2000, CMC, or CIE94.

Index	Layer type	Shape	Activation
0	Input	$I = I^{(0)} = 2, J^{(0)} = 45$	tanh
[1, 2]	Residual	$I^{(1)} = J^{(2)} = 45$ $I^{(2)} = J^{(1)} = 15$	tanh
3	Output	$I^{(3)} = 45, J = J^{(3)} = d - 1$	id

5.1 | Layout of neural network color space transformations

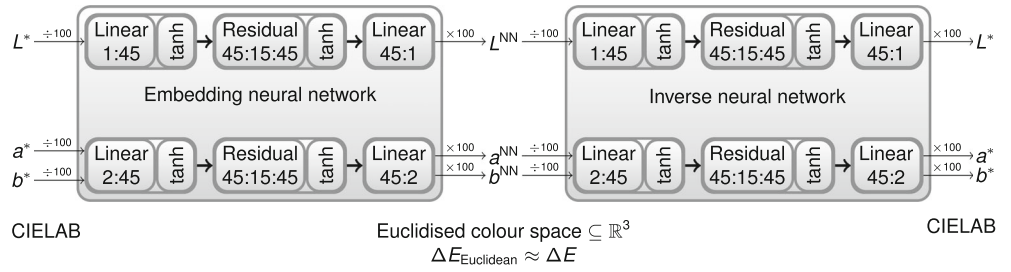
For the sake of comparative evaluation and in accordance with previous contributions,^{12,17} the L-coordinate and the ab-plane of the CIELAB color space are treated independently and processed through two separate (sub-)neural networks with the same architecture.³ Moreover, since the standard methods for initializing the neural network parameters rely on the assumption that the magnitudes of the entries of the input vector are not much larger than 1, as a normalization step, the input to each neural network color space transformation is divided by 100 and, correspondingly, the output is multiplied by 100 (coordinate-wise).

The (sub-)neural network for embedding the ab-plane of the CIELAB color space into \mathbb{R}^{d-1} employs a single residual hidden layer of size 45 : 15 : 45, as presented in Table 1. The corresponding inverse neural network transformation uses the same layout, apart from flipped numbers of input and output channels. Analogously, the embedding and inverse transformations associated with the L-coordinate of the CIELAB color space employ the same layouts as those for the ab-plane, except that the numbers of input and output channels are all set to 1.⁴ The embedding and inverse neural network transformations in case of $d = 3$ are visualized in Figure 6, with thin arrows passing scalar values and thick arrows passing vectors.⁵

5.2 | Training configuration

The mini-batch size for running Subroutine R1 in Section 3.5 is determined through the total number of color centers sampled from the reference region R per mini-batch times the total number of directions sampled from $S^2 \subseteq \mathbb{R}^3$ per color center (recall training data generation steps in Sections 3.3 and 4), which are set to $N(\iota) := 1000$, $K(\iota) := 100$. For running Subroutine R2 in Section 3.5, the mini-batch size is the total number of color samples generated on R per mini-batch, which is set to $N(\iota) := 100000$.

FIGURE 6 Embedding and inverse neural network transformations for CIELAB-based examples in case of $d = 3$.



Among the considered example scenarios, the ones with $d := 3$ and $[r_{\min}, r_{\max}] := [10^{-3}, 5]$ are the most relevant, for which the embedding and inverse neural network transformations are jointly trained using the proposed regularization measures. As a preliminary step for tuning the hyperparameters $(\gamma, \tilde{\gamma}, \beta)$, $\#R1(i)$, and $\#R2(i)$ in Algorithm 1, the embedding neural network is first trained with Subroutine R1 alone, so as to set a target performance and roughly estimate the total number of forward iterations, $\sum_i \#R1(i)$, needed over the entire training course with regularization. As a candidate for the forward-to-inverse ratio, $[\#R1(i) : \#R2(i)]$, a base number of rotations per training block, I , a fixed number of forward iterations per rotation, $\#R1(i)$, and a block-wise linearly increasing number of inverse iterations $\#R2(i)$ are chosen. Here, the linear increase aims to compensate for the typically convex decreasing trend of the inverse training loss. In total, 15 training blocks of $I := 10$ rotations, with rotation lengths ($\#R1(i) := 4000$, $\#R2(i) := 16000(1 + \lfloor i/10 \rfloor)$) for $i < 150$, are executed at learning rates $(\gamma := \tilde{\gamma} := 10^{-5}, \beta := 10^{-8})$ for determining an embedding color space transformation in each setting. As accuracy requirement for the inverse transformation, the total transformation accuracy evaluated on the DE2000 color difference formula in Reference 12 is taken as baseline.⁶ In 5 of the 6 considered primary scenarios, this accuracy of the inverse transformation is achieved while terminating the last training block, with the only exception being the scenario where the mean squared relative isometric disagreement with the CMC color difference formula is minimized, for which a follow-up inverse training with Subroutine R2 (where $\beta := 0$) at a learning rate of $\tilde{\gamma} := 10^{-5}$ is run until the stop criterion is fulfilled. To give an impression of the actual duration of the training process: on a single NVIDIA Quadro RTX 5000 graphics processing unit (GPU), the time required for running 150 training blocks for the DE2000 color difference formula with respect to the relative isometric disagreement amounts to about 61.5 h in total, with about 1 h and 43 min spent on the forward training and slightly under 60 h spent on the inverse training. In particular, the most expensive part is the training of the inverse transformation, which reflects the training schedule introduced above (recall the configured forward-inverse-ratio $[\#R1(i) : \#R2(i)]$ for $i < 150$).

Since the inverse training is extremely time-consuming, for the remaining supplementary example scenarios where $[r_{\min}, r_{\max}] := [5, 20]$ or $d = 4, 5, 6$, only the embedding neural network is trained and all regularization measures are omitted (visual inspection of injectivity indicates no invertibility issue). Since those scenarios are considered for rather experimental purposes, no extra tuning is performed therefor and Subroutine R1 with a learning rate of $\gamma := 10^{-5}$ is simply run for 600000 or 2400000 iterations in case of $[r_{\min}, r_{\max}] := [5, 20]$ or $d = 4, 5, 6$, respectively.

6 | EVALUATION RESULTS

In this section, the proposed machine learning method for color space Euclidization is evaluated on the example scenarios introduced in Section 4 and compared to previous contributions. In each example scenario, 2 million randomly generated test color pairs are used for evaluation (recall generation procedure described in Section 4). The performance metrics considered while evaluating the embedding transformations are the STRESS index (recall (12)) and the (symmetric) relative isometric disagreement (recall (3)) evaluated statistically in its mean, maximum, and standard deviation. To assess the accuracy of the corresponding inverse approximations, the point-wise Euclidean distances between the identity mapping and the composition of the embedding and inverse neural network transformations are evaluated statistically in the mean, maximum, and standard deviation; as noted in Section 5.2, implied by the stop criterion for the inverse training, these indices are below 0.0221, 0.3939, and 0.0176, respectively, and will not be further specified in the sequel.

6.1 | Results for DE2000 color difference formula

The DE2000 color difference formula is implemented according to Reference 28 where $k_L := k_C := k_H := 1$. The reference methods considered for comparative evaluation on this formula are the multigrid optimization method¹² and the multidimensional scaling techniques,¹⁷ referred

TABLE 2 Performance comparison when embedding CIELAB into \mathbb{R}^3 w.r.t. DE2000 for small color differences.

Method	STRESS	Relative isometric disagreement		
		Mean	Max	σ
Multigrid	0.0326	0.0274	1.7280	0.0488
MDS	0.0295	0.0235	0.2627	0.0307
NN mrd	0.0225	0.0116	0.4421	0.0199
NN mse	0.0181	0.0137	0.8779	0.0251

Note: Bold values are the minima of the corresponding columns.

to as Multigrid and MDS, respectively. The neural network color space embeddings trained for minimizing either the mean squared relative isometric disagreement or the mean squared absolute isometric disagreement are referred to as NN mrd or NN mse, respectively.

6.1.1 | Comparative evaluation on small color differences

The results of the comparative evaluation for small color differences are presented in Table 2. Compared to the multigrid optimization method (Multigrid) which is based on minimization of the mean squared absolute isometric disagreement, the neural network transformation trained with the same optimization criterion (NN mse) exhibits a 44% reduction in the STRESS index (recall correspondence elaborated in Section 3.4.2). Compared to the multidimensional scaling techniques (MDS) which employ the (asymmetric) relative isometric disagreement as key disagreement measure for optimization, the embedding neural network trained with a similar criterion, NN mrd, performs better in the mean and standard deviation of the relative isometric disagreement, with the corresponding second moment (i.e., $\text{mean}^2 + \sigma^2$) being more than halved, but almost doubles the maximum relative isometric disagreement. The excellence of the MDS method in lowering the maximum relative isometric disagreement relies in part on this metric being included in the objective function and minimized directly through the optimization algorithm therein. More generally, both of the reference methods (Multigrid and MDS) rely on local optimization while looping through meshes defined on the original color space, which implies a greater potential for producing better local approximation results. In contrast, the present machine learning method is based on minimization of average loss over the entire color space and thereby has less control over local features of the resulting color space transformation. In fact, except for the

maximum relative isometric disagreement, all considered performance metrics are average-based disagreement measures, in respect of which the proposed approach turns out to be more effective than the reference methods, regardless of the choice between the relative and absolute disagreement measures being employed in the training criterion.

Among the proposed embedding neural network transformations, NN mrd and NN mse, the former is superior to the latter in the second moment of the relative isometric disagreement (i.e., $\text{mean}^2 + \sigma^2$), and vice versa in the STRESS index, where the respective performance advantages amount to 35% and 20%. Figure 7 displays the images of a (rectangular) grid in the ab-plane of CIELAB under the respective neural network transformations; here and in the sequel, the colored region in the plot is the image of the intersection between the corresponding region in CIELAB and the sRGB gamut for constant lightness of $L^* = 50$.^{7,8} Comparing both plots, the geometry of the two transformations differs visibly; in particular, the transformation optimized with respect to the mean squared relative disagreement (NN mrd) exhibits a stronger shearing effect at the center of the ab-plane. Figure 8 shows the local maximum relative isometric disagreement of the embedding neural networks evaluated on the ab-plane of CIELAB; in the images, each pixel corresponds to a color center and the local maximum relative isometric disagreement is evaluated on color difference vectors around this center.⁹ From Figure 8A, in the optimization with respect to the mean squared relative disagreement (NN mrd), the highest relative disagreement values are located near the gray axis on the one hand and in the region between hue angles 270° and 295° on the other. From Figure 8B, in the optimization with respect to the mean squared absolute disagreement (NN mse), the local maximum relative disagreement is particularly large around hue angle 275° , which is consistent with the results in Reference 12 and the discussion on (approximate) Gaussian curvature provided therein. The spatial distribution of the isometric disagreement is further illustrated through the unity color difference ellipses in the transformed ab-plane as depicted in Figure 9, where some skew is still present in the Euclidized coordinates and the largest distortions of the unity circle are located in accordance with the highest isometric disagreement values.

6.1.2 | Comparative evaluation on small color differences

For experimental purposes, color differences in the range of 5 to 20 CIELAB-units are also considered for

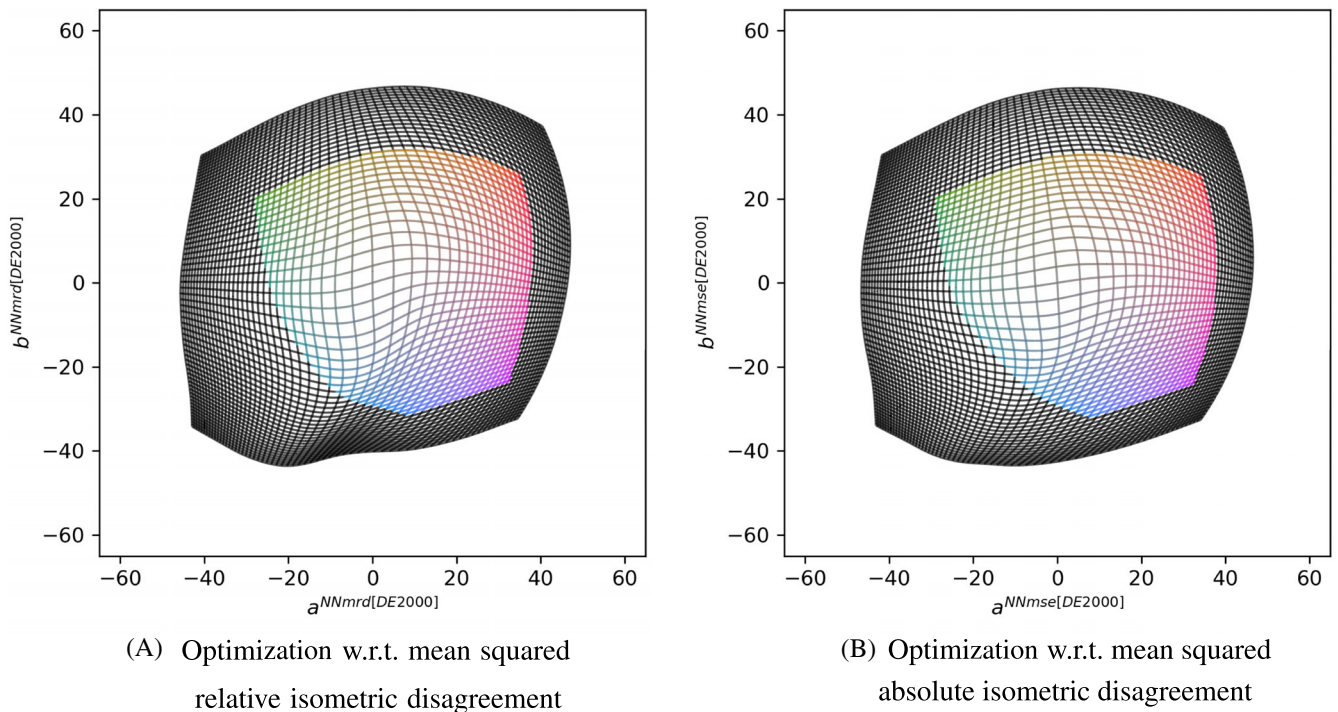


FIGURE 7 Images of a rectangular grid in the ab-plane of CIELAB under neural network color space transformations trained according to the DE2000 color difference formula.

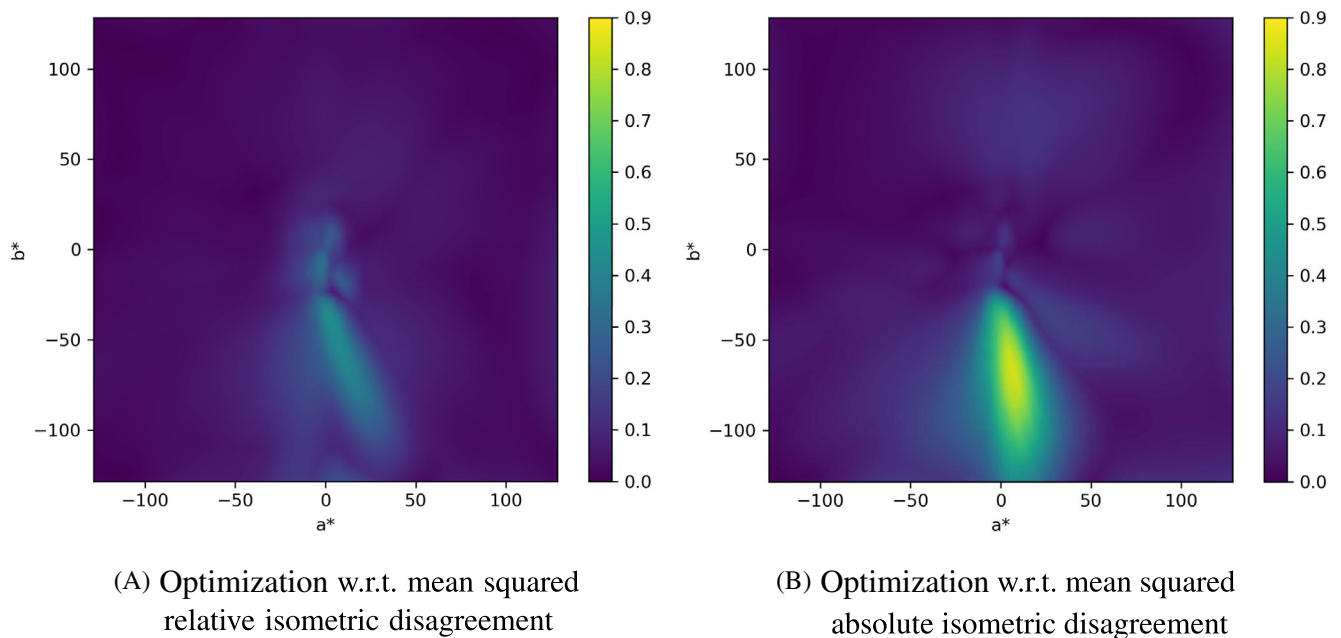


FIGURE 8 Local maximum relative isometric disagreement of neural network color space embeddings w.r.t. DE2000, evaluated on the ab-plane of CIELAB.

evaluation. The main aim here is to explore the feasibility of the present approach for Euclidization with respect to large color differences. The reason for the DE2000 color difference formula being taken as reference law for such experimentation is the comparability of the outcome with the preceding results for small color differences, even

though the efficacy of this formula in predicting a higher range of visual differences is yet to be verified. After the first 600000 iterations of training with respect to either the relative or the absolute disagreement measure, the resulting neural network color space transformation delivers a STRESS index of 0.0231 or 0.0189, respectively,

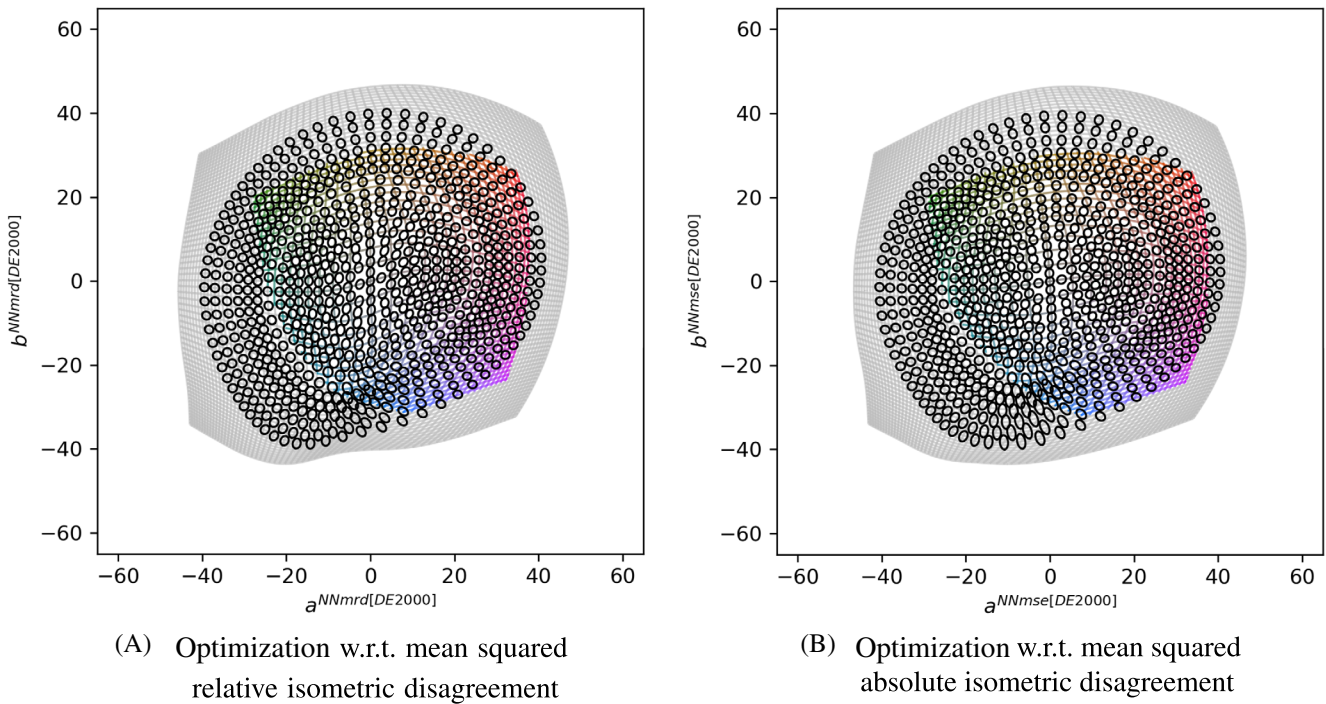


FIGURE 9 Unity color difference ellipses in the embedded ab -plane associated with neural network color space transformations trained for the DE2000 color difference formula.

and the mean, maximum, and standard deviation of the relative isometric disagreement amount to $(0.0122, 0.4127, 0.0197)$ or $(0.0144, 0.8329, 0.0248)$, respectively. These disagreement values are similar to the preceding ones in Table 2 evaluated on small color differences, with no significant decline of performance being identified in any of the considered metrics. This in particular demonstrates that the proposed machine learning method is able to handle color differences that are significantly away from zero, as opposed to the other two numerical methods in References 12,17 for which a direct adaptation to large color differences is nonobvious.

6.1.3 | Dimensional analysis

Another advantage of employing artificial neural networks for approximating color space transformations is the possibility of a straightforward alteration in the dimension of the input and output vectors. Making use of this feature, embeddings with higher-dimensional codomain for $d = 4, 5, 6$ are explored where the corresponding neural networks are endowed with the respective numbers of output channels (recall Table 1). Actually, compared to the case of $d = 3$, the higher-dimensional embedding problem is more complex in terms of degrees of freedom being required; for $d \geq 4$, deeper neural networks, for instance, those endowed with 4 residual

TABLE 3 Evaluation results for embedding CIELAB into \mathbb{R}^d w.r.t. DE2000, where $d = 3, 4, 5, 6$.

d	STRESS	Relative isometric disagreement		
		Mean	Max	σ
3	0.0225	0.0116	0.4421	0.0199
4	0.0032	0.0009	0.1351	0.0021
5	0.0023	0.0006	0.0978	0.0014
6	0.0019	0.0005	0.0614	0.0012

hidden layers, exhibit better performance and faster convergence during experimentation. However, for the sake of comparability with the preceding results for $d = 3$, the same inner layout of the embedding neural network is kept through all considered dimensions and, as a compensation for the lower capacity, the total number of training iterations for $d \geq 4$ is set to be four times as large as that for $d = 3$.

The evaluation results of the dimensional analysis are provided in Table 3. Similarly to the results in Reference 17, when including one more dimension in the codomain of the embedding, that is, transitioning from $d = 3$ to $d = 4$, the performance of the resulting color space transformation is improved significantly in all considered disagreement metrics. This effect diminishes when further increasing the output dimension to $d = 5$, and there is

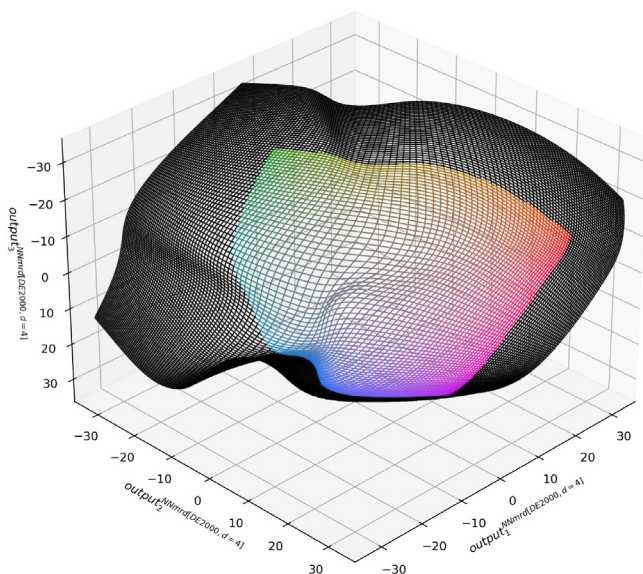


FIGURE 10 Image of a rectangular grid in the ab -plane of CIELAB embedded into \mathbb{R}^3 via a neural network color space transformation trained according to the DE2000 color difference formula.

only marginal performance improvement when transitioning from $d = 5$ to $d = 6$. Compared to the evaluation results for $d = 4$ in Reference 17, tab. 1, the neural network color space transformation exhibits significantly better performance than the transformation obtained through the multidimensional scaling techniques. As visualized in Figure 10 and in comparison with Reference 17, fig. 3, the present higher-dimensional embedding transformation exhibits a smoother appearance with less folding of the embedded surface. This results from the infinitely differentiable nature of the neural network transformation as well as the fewer degrees of freedom inherent in the present global optimization approach (i.e., minimization of average loss over the entire color space) compared to the local optimization method in Reference 17.

6.2 | Results for CMC color difference formula

The implementation of the CMC(l:c) color difference formula is based on Reference 6 with $l := c := 1$. For the sake of symmetry in the sense of the defining properties of a metric and as previously also implemented in Reference 12, the weights S_L , S_C , and S_H appearing in the formula are computed with respect to the mean coordinates of each color pair. The reference method considered in the comparative evaluation for this formula is the multigrid optimization method¹² (Multigrid). The neural network

TABLE 4 Performance comparison when embedding CIELAB into \mathbb{R}^3 w.r.t. CMC for small color differences.

Method	STRESS	Relative isometric disagreement		
		Mean	Max	σ
Multigrid	0.0239	0.0210	0.4395	0.0311
NN mrd	0.0108	0.0062	0.2469	0.0097
NN mse	0.0098	0.0071	0.1831	0.0106

Note: Bold values are the minima of the corresponding columns.

color space transformations trained with respect to the mean squared relative or absolute isometric disagreement are again referred to as NN mrd or NN mse, respectively.

The evaluation results are presented in Table 4. In all considered performance metrics, both of the proposed neural network color space embeddings exhibit significant performance advantages compared to the reference method, with most of the reference disagreement values being more than halved. It should be noted that the exact manner in which the symmetrization of the CMC color difference formula is implemented may affect the final results; to the best of the authors' knowledge, this is not explicitly specified in any standards or reference implementation. In the present work, the arithmetic means of the lightness, chroma, and hue coordinates of each color pair are computed as in the implementation of the DE2000 formula detailed in Reference 28, which in particular includes an ambiguity resolution method for the hue angles and the hue angle difference. Table 5 demonstrates the effect of the symmetrization, where the neural network transformations associated with the two considered optimization criteria, NN mse and NN mrd, are trained with either the symmetrized or the original (asymmetric) CMC color difference formula as ground truth and then cross-evaluated on the symmetrized or the original (asymmetric) formula through the STRESS index. Overall, the performance of the neural network transformations degrades by a factor of about 2.5 when evaluated against the asymmetric CMC formula (last line vs. second-last line of table). In fact, without prior symmetrization, the asymmetry of the original CMC formula severely conflicts with the inherently symmetric nature of the Euclidean distances resulting from the neural network color space transformations, which leads to large increase in the STRESS indices that overshadows the more subtle differences resulting from the distinction between the two training loss functions. On the other hand, comparing the third column with the second column, or the fifth column with the fourth column of the table, training the neural networks with the original asymmetric CMC formula instead of the symmetrized version leads to hardly any performance improvement

STRESS indices

Method	NN mse		NN mrd	
	Symm.	Asymm.	Symm.	Asymm.
ΔE during training				
Tested on symmetrized ΔE	0.0098	0.0109	0.0108	0.0119
Tested on asymmetric ΔE	0.0263	0.0261	0.0266	0.0267

TABLE 5 Effect of symmetrizing the CMC color difference formula.

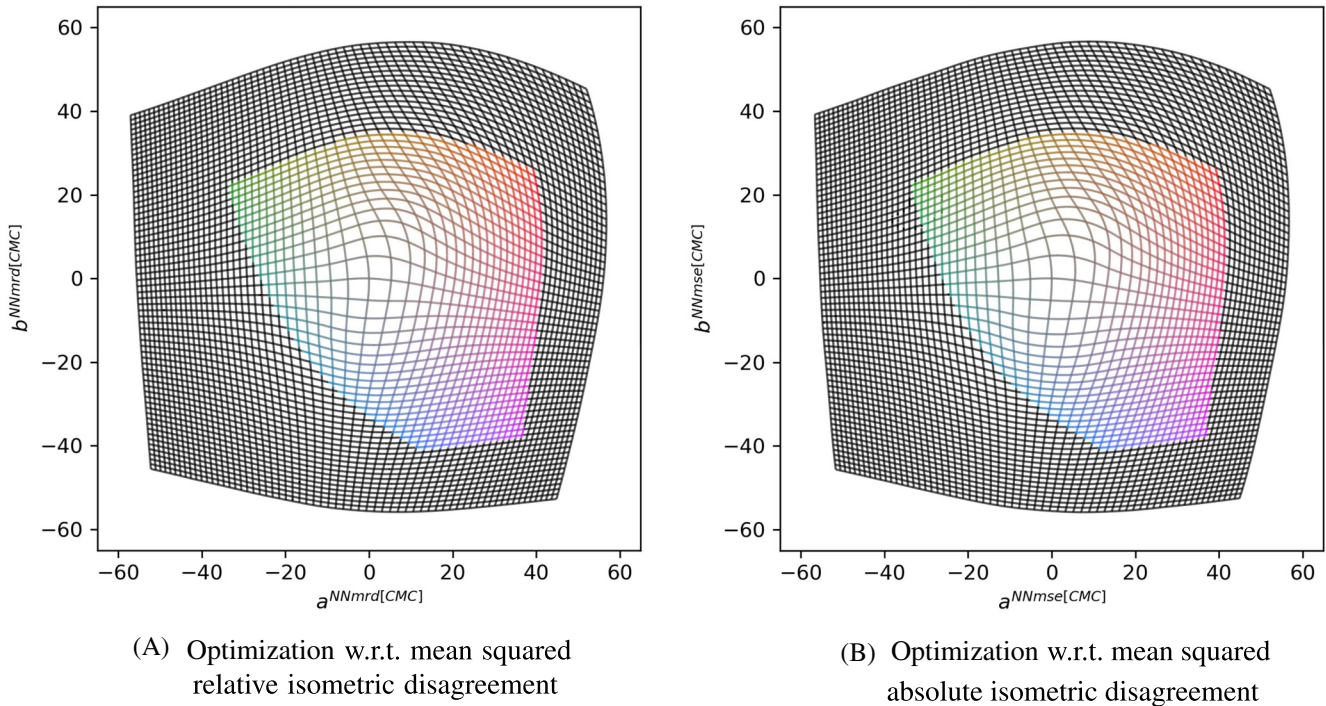


FIGURE 11 Images of a rectangular grid in the ab-plane of CIELAB under neural network color space transformations trained according to the CMC color difference formula.

(or even performance decline) when evaluated on the original asymmetric formula, but significantly increases the isometric disagreement when evaluated on the symmetrized formula. From the above considerations and in view of the practical advantages of using a symmetric description of color differences, the symmetrized version of the CMC color difference formula is preferred over the original asymmetric one while training the corresponding neural network color space embeddings.

When comparing the proposed neural network color space transformations NN mrd and NN mse among one another, the former exhibits a 19% lower second moment of the relative disagreement (i.e., $\text{mean}^2 + \sigma^2$) and the latter produces a 9% lower STRESS index. Compared to the preceding results for the DE2000 color difference formula, smaller performance differences between the two neural network transformations are observed across all considered metrics, which may be attributed to the lower complexity of the CMC color difference formula

expressed in terms of its deviation from the distance function of a Riemannian manifold with zero curvature. The diminished dependency on the optimization criterion is also reflected in the geometry of the respective transformations, as depicted in Figure 11,¹⁰ where only a slightly larger magnification of the center area in the transformation optimized with respect to the mean squared absolute disagreement (NN mse) is noticeable. The impact of the optimization criterion is more visible in the spatial distribution of the disagreement values within the ab-plane of CIELAB displayed in Figure 12. Similarly to the Euclidization according to the DE2000 color difference formula, optimization with respect to the mean squared relative isometric disagreement (NN mrd) results in larger relative disagreement values being more concentrated near the gray axis. In contrast, in the optimization with respect to the mean squared absolute isometric disagreement (NN mse), the local maximum relative disagreement becomes slightly more uniformly

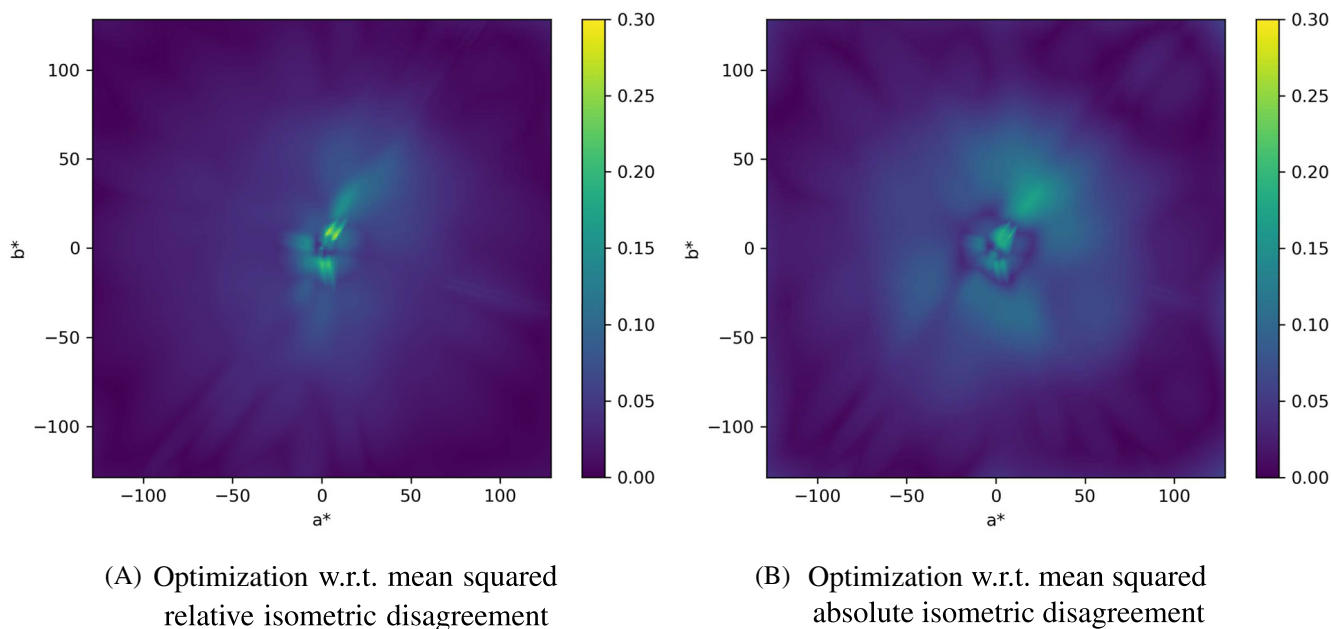


FIGURE 12 Local maximum relative isometric disagreement of neural network color space embeddings w.r.t. CMC, evaluated on the ab-plane of CIELAB.

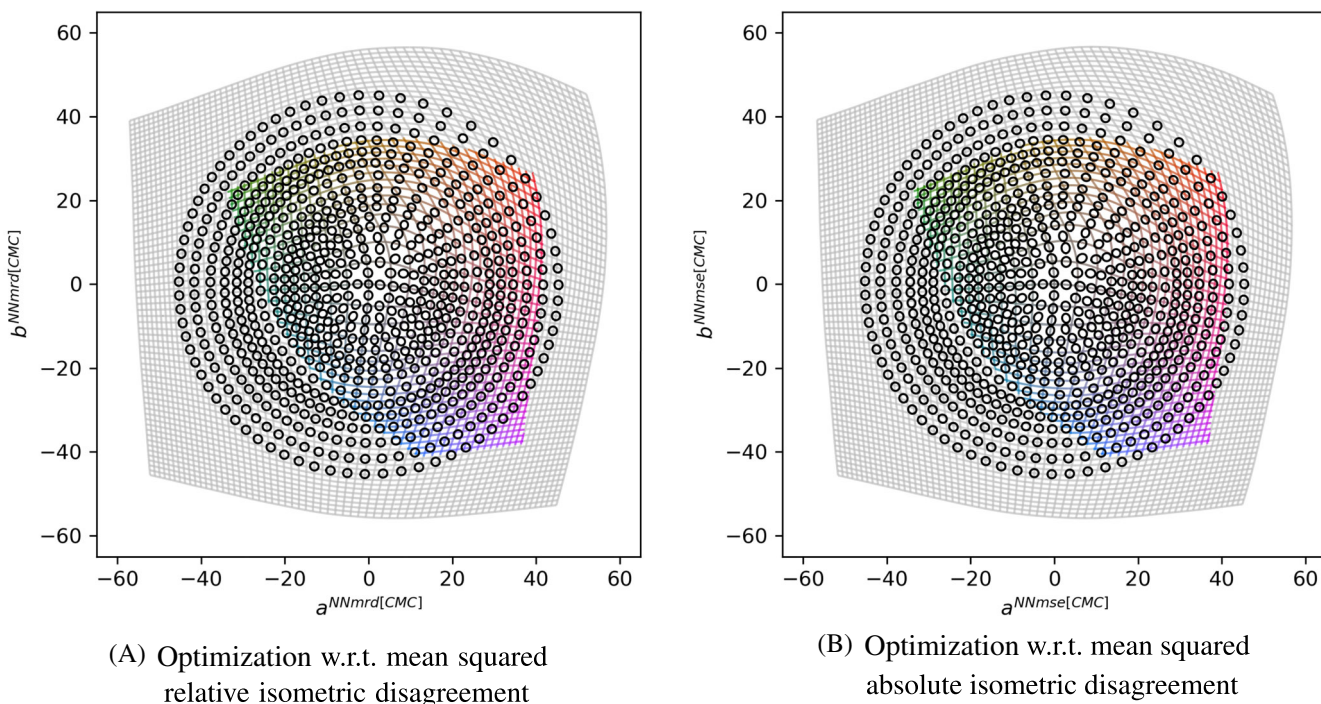


FIGURE 13 Unity color difference ellipses in the embedded ab-plane associated with neural network color space transformations trained for the CMC color difference formula.

distributed and the area with higher disagreement values around the center is slightly further spread out in the ab-plane. On top of the region near the gray axis, both optimizations also exhibit larger relative disagreement around hue angle 45° , which is consistent with the

results in Reference 12. As shown in Figure 13, the unity color difference ellipses in both of the transformed ab-planes appear very uniform with little skew, where the few ellipses exhibiting visible eccentricity are located in the regions of the largest isometric disagreement values.

6.3 | Results for CIE94 color difference formula

The implementation of the CIE94 color difference formula is based on Reference 8 with $k_L := k_C := k_H := 1$, $K_1 := 0.045$, and $K_2 := 0.015$. Again, for the sake of symmetry in the sense of the defining properties of a metric, the weights S_C and S_H in the formula are determined through the arithmetic mean of the chroma coordinates of each color pair. For this color difference formula, the reference methods evaluated for comparison are the multigrid optimization method¹² (Multigrid) and the analytical method in Reference 15. For comparative purposes, the original (follow-up) convex optimization for minimizing the maximum relative isometric disagreement in Reference 15 is adapted to the present setting by minimizing instead the mean squared relative or absolute isometric disagreement in accordance with Section 3.4 through gradient descent, and the resulting embedding transformations (determined by the correspondingly adjusted constants P and Q in Reference 15) are referred to as Analytic mrd or Analytic mse, respectively. Note that this adaptation of the method does not lead to a degradation of the overall performance, but on the contrary, except for the maximum relative disagreement of Analytic mrd, all considered performance metrics are improved in the present evaluation. Again, the neural network color space transformations optimized with respect to the mean squared relative or absolute isometric disagreement are referred to as NN mrd or NN mse, respectively.

6.3.1 | Comparative evaluation on small color differences

The evaluation results for small color differences are summarized in Table 6. Compared to the multigrid optimization method (Multigrid) which is based on minimizing the mean squared absolute isometric disagreement, the embedding

TABLE 6 Performance comparison when embedding CIELAB into \mathbb{R}^3 w.r.t. CIE94 for small color differences.

Method	STRESS	Relative isometric disagreement		
		Mean	Max	σ
Multigrid	0.0113	0.0077	0.1447	0.0126
Analytic mrd	0.0083	0.0043	0.1276	0.0074
Analytic mse	0.0076	0.0058	0.0890	0.0077
NN mrd	0.0074	0.0040	0.1185	0.0069
NN mse	0.0069	0.0043	0.0872	0.0073

Note: Bold values are the minima of the corresponding columns.

neural network trained with the same optimization criterion (NN mse) delivers a 39% lower STRESS index. Compared to the analytical method, the neural network color space transformations optimized with respect to the relative or absolute disagreement measure (NN mrd/mse) slightly outperform their respective analytical counterparts (Analytic mrd/mse) in the corresponding reference indices (by about 10%). Interestingly, in contrast to the preceding results for the DE2000 color difference formula, changing from the absolute to the relative disagreement measure in the objective function seems to cause an increase in the maximum relative isometric disagreement, which is observed for both the machine learning and the analytical approaches. Overall, the neural network color space embeddings again exhibit lower average isometric disagreement than the transformations of the reference methods, which is reflected in their corresponding STRESS indices and means and standard deviations of the relative isometric disagreement, irrespective of the optimization criterion being employed during training. In addition, Table 7 provides the evaluation of the symmetrization effect with respect to the CIE94 color difference formula, where the neural network embeddings associated with the absolute or relative disagreement measure, NN mse and NN mrd, are trained with either the symmetrized or the original asymmetric version of the formula as ground truth and subsequently cross-evaluated on these two formulas through the STRESS index. As expected, evaluating the nearly isometric embeddings on the original asymmetric formula degrades the performance in general (last line vs. second-last line). However, the effect is much less noticeable compared to the case of CMC, showing a performance decline of at most 20%. Moreover, training with the original asymmetric formula instead of the symmetrized version makes hardly any difference in performance, no matter which version of the formula is taken as reference for evaluation (compare the second with the third column, or the fourth with the fifth column of the table).

With CIE94 being the simplest among all considered color difference formulas, the lowest isometric disagreement values are observed for this formula and the performance variation across the neural network color space transformations trained with different optimization criteria dilutes further in value (with less than 7% difference in all average-based indices). As illustrated in Figure 14,¹¹ the dependency of the embedding transformations on the choice of optimization criterion is hardly discernible in respect of their geometry; when comparing the two images very closely, a slightly more stretched central area is identified in the transformation minimizing the mean squared relative disagreement (NN mrd). The difference becomes more distinct when comparing the respective

TABLE 7 Effect of symmetrizing the CIE94 color difference formula.

STRESS indices				
Method	NN mse		NN mrd	
	Symm.	Asymm.	Symm.	Asymm.
ΔE during training				
Tested on symmetrized ΔE	0.0069	0.0070	0.0074	0.0074
Tested on asymmetric ΔE	0.0083	0.0083	0.0087	0.0087

FIGURE 14 Images of a rectangular grid in the ab-plane of CIELAB under neural network color space transformations trained according to the CIE94 color difference formula.

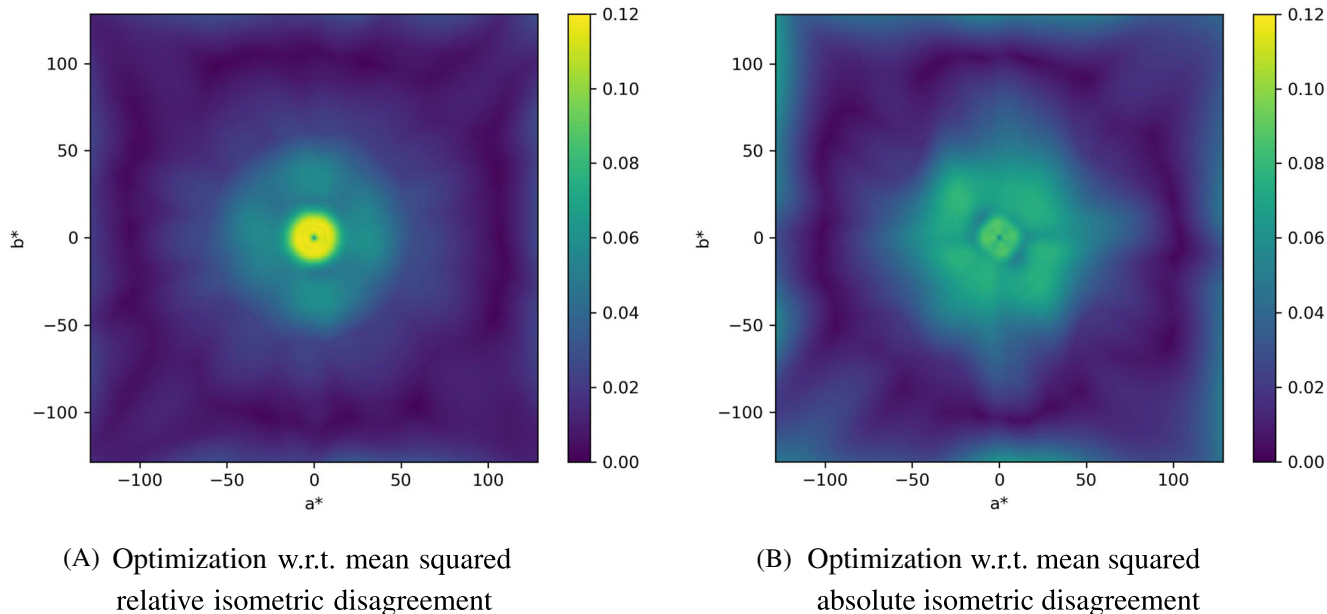
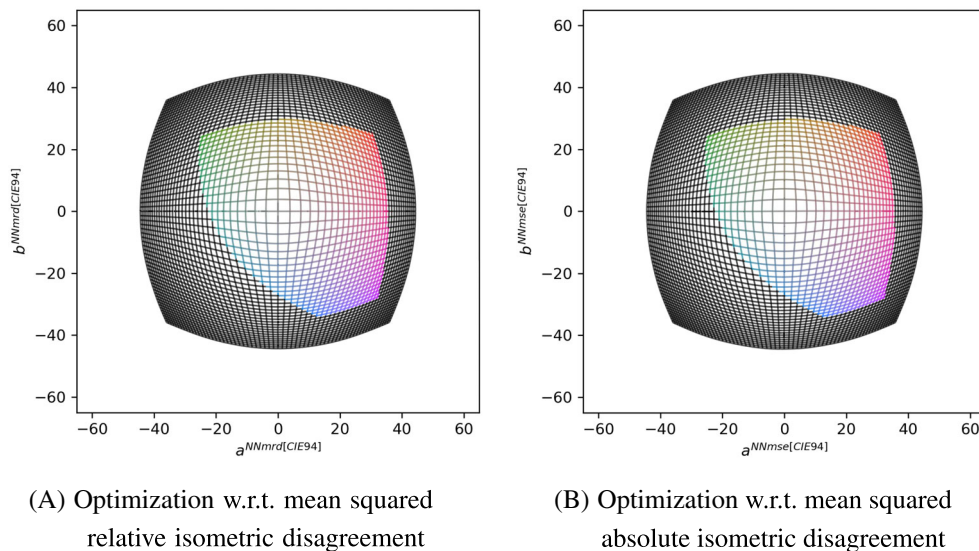
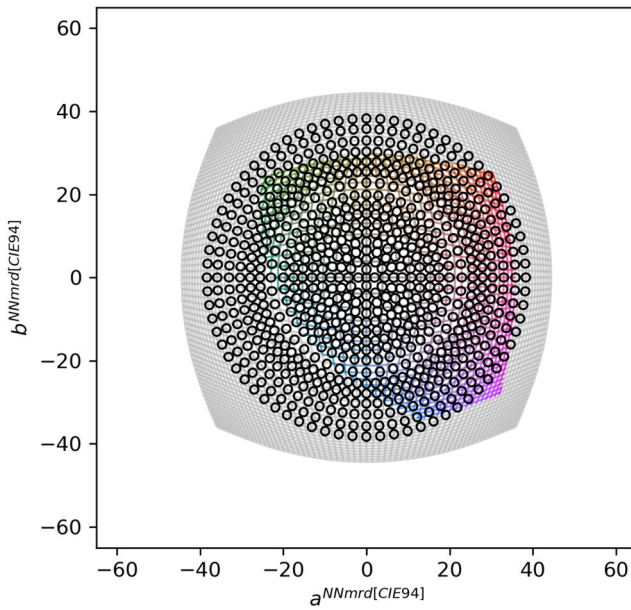


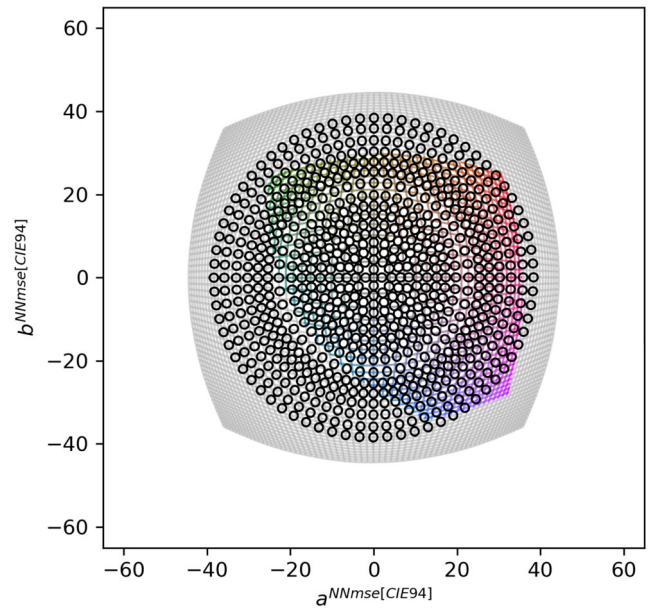
FIGURE 15 Local maximum relative isometric disagreement of neural network color space embeddings w.r.t. CIE94, evaluated on the ab-plane of CIELAB.

local maximum relative disagreement displayed in Figure 15, where the optimization with respect to the mean squared absolute disagreement (NN mse) spreads the highest local maximum disagreement values across a larger area around the gray axis, leading to a slightly

lower global maximum value compared to the alternative (NN mrd). As expected, for both embedding neural networks, the rotational symmetry of the CIE94 color difference formula translates to a high degree of rotational symmetry in both the geometry of the transformation



(A) Optimization w.r.t. mean squared relative isometric disagreement



(B) Optimization w.r.t. mean squared absolute isometric disagreement

FIGURE 16 Unity color difference ellipses in the embedded ab-plane associated with neural network color space transformations trained for the CIE94 color difference formula.

and the spatial distribution of the relative isometric disagreement. The unity color difference ellipses in both of the transformed ab-planes presented in Figure 16 appear nearly perfectly circular, which indicates a high agreement of the Euclidean metric on the transformed spaces with the underlying color difference formula.

6.3.2 | Evaluation on large color differences

Since the CIE94 color difference formula exhibits to a certain extent improvement over the Euclidean metric in CIELAB space (CIE76) in predicting large-scale visual differences (as opposed to the CMC formula, cf. Reference 29), as one more example for demonstrating the broad applicability of the proposed machine learning method, Euclidization with respect to color differences from 5 to 20 CIELAB-units is also evaluated on this formula. After the first 600 000 iterations of training with respect to the mean squared relative or absolute isometric disagreement, the STRESS index and the mean, maximum, and standard deviation of the relative isometric disagreement of the resulting neural network color space transformations amount to 0.0074, (0.0039, 0.1203, 0.0070) or 0.0070, (0.0043, 0.1048, 0.0074), respectively. As with the preceding evaluation on the DE2000 color difference formula, the results for small and large color differences herein are similar (compare Table 6); in particular, no significant deterioration in any of the considered

disagreement indices is identified in the evaluation results for large color differences. This again verifies that the proposed approach is not limited to small color differences.

7 | DISCUSSION

Subsequently, four subjects are discussed: the computational effort required during the inference stage of the proposed neural network approach (cf. Section 7.1), the total performance of Euclidized color difference systems in terms of their perceptual uniformity evaluated on experimental visual data (cf. Section 7.2), the feasibility of adapting the present method to the task of color space modeling with experimental visual data (cf. Section 7.3), and comparison as well as potential intersections between the present work and a selection of machine learning approaches to perception-based image processing issues (cf. Section 7.4).

7.1 | Computational effort

A typical issue with machine learning based methods is the relatively large amount of computational resources required to perform the necessary computations. In the present work, particular care is taken to devise efficient neural networks, so as to avoid using an excessive

amount of operations (recall discussion on residual layers in Section 3.1).

According to the layout presented in Section 5.1 (Table 1 and Figure 6), in case of $d=3$, transforming a single point in the original color space into its coordinates in the embedded color space through the proposed neural network color space transformation implies a computational burden of 2970 multiplications, 3060 additions, and 210 evaluations of the hyperbolic tangent activation function. Here, one third of the multiplications associated with the densely connected alternative are saved through the architecture with residual connections. While this transformation is still computationally more expensive than traditional color space conversions, it is not prohibitively costly when performed on modern graphics processing hardware. In this context, it should also be noted that the computational effort of the neural network color space transformations herein is much lower than that of some modern image enhancement algorithms such as machine learning based super resolution methods (cf. e.g., Reference 30). For reference, when performed on the same hardware as employed for training the neural network transformations (NVIDIA Quadro RTX 5000 GPU), the time for converting the color coordinates of an image in four common resolutions amounts to 5.4ms for an image in VGA resolution, 16.2ms for an HD (720 p) image, 36.8ms for a Full HD (1080 p) image, and 149.3ms for a 4K UHD image. It should be noted that, at this stage, no attempt at optimizing the algorithm for reduced conversion time (e.g., through the use of lookup tables or quantization) has been made.

In total, the neural network employed in this exposition of the method is parametrized with 3183 free floating point parameters (for $d=3$). The amount of memory required for storing these parameters is less than that required for the storage of the look-up table employed in Reference 12 or the pairs of grid points defining the transformation developed in Reference 17.

Depending on the color difference formula as well as other factors such as requirements on accuracy or training speed, a modification of the neural network layout decreasing the computational effort and memory requirements may be feasible. In fact, embedding the original color space into \mathbb{R}^3 is a much easier task compared to the corresponding inverse approximation or the problem of higher-dimensional embedding for $d \geq 4$ in terms of network capacity being required; the layout in Table 1 is chosen mainly in consideration of the speed and stability of the inverse training and is therefore rather generous in size in respect of the actual Euclidization problem. For instance, experimentation reveals that, if a 5% higher mean squared relative isometric disagreement with the DE2000 color difference formula is

acceptable, it suffices to endow the corresponding embedding neural network with only one third of the existing neurons, which then requires about 20% of the original computational effort. Further reduction of both the computational effort and the size of the parameter storage may be achieved through quantization and the use of fixed point arithmetic.

7.2 | Perceptual uniformity of Euclidized color difference system

As noted in the introduction Section 1, functioning as a new color space, the Euclidized color difference system endowed with the Euclidean metric is expected to exhibit similar perceptual uniformity to that of the underlying reference color difference formula, as long as the isometric disagreement in the Euclidization step is insignificant. To demonstrate this, the currently CIE-recommended DE2000 color difference formula is taken as reference and the four original experimental visual datasets employed for its derivation, BFD-P,³¹ Leeds,³² RIT-DuPont,³³ and Witt,³⁴ are considered for evaluation¹² of the corresponding two Euclidean spaces resulting from the neural network nearly-isometric embeddings, referred to as NNmrd [DE2000] and NNmse[DE2000] in the sequel (recall Section 6.1 for analogous notations). In accordance with Reference 35, the STRESS between the visual and measured color differences (ΔV_i vs. ΔE_i) is employed as performance metric for assessing the perceptual uniformity of each color difference system and the statistical significance of the performance deviation from the reference color difference formula is evaluated through the two-tailed F -test (i.e., the F -statistic is determined through the ratio of two squared STRESS indices, cf. Reference 20 for more details). The evaluated STRESS indices and F -statistics are given in Table 8 where values associated with the reference color difference formula, DE2000, are highlighted in bold. From the STRESS evaluation (upper half of table), the performance of both Euclidized color difference systems, NNmrd[DE2000] and NNmse [DE2000], is close to that of the DE2000 color difference formula. Across various datasets, the STRESS values of both embedded spaces fluctuate slightly around the reference values associated with DE2000, with the maximum relative performance gain below 9% identified in the RIT-DuPont dataset and the maximum relative performance decline below 3% identified in the Leeds dataset. According to the results of the F -test (lower half of table), all evaluated F -statistics fall within the mid-range of the corresponding 95%-confidence intervals, which indicates that there is no statistically significant difference between the perceptual uniformity of NNmrd

TABLE 8 Perceptual uniformity of the Euclidean color spaces resulting from the neural network embeddings w.r.t. the DE2000 color difference formula evaluated on experimental visual data.

	BFD-P	Leeds	RIT-DuPont	Witt
STRESS [DE2000]	0.2955	0.1925	0.1947	0.3022
STRESS [NNmrd[DE2000]]	0.2948	0.1975	0.1774	0.3020
STRESS [NNmse[DE2000]]	0.2938	0.1971	0.1850	0.3049
95%-confidence interval for <i>F</i>-statistic w.r.t. DE2000	[0.937, 1.067]	[0.799, 1.252]	[0.729, 1.372]	[0.825, 1.212]
<i>F</i> -statistic [NNmrd[DE2000]]	0.9976	1.0260	0.9110	0.9994
<i>F</i> -statistic [NNmse[DE2000]]	0.9941	1.0238	0.9501	1.0088

[DE2000] or NNmse[DE2000] and that of the original DE2000 color difference system. For an extensive performance comparison across various color difference systems as well as an indepth discussion on proper statistical evaluation of performance discrepancy in respect of perceptual uniformity, the reader is referred to Reference 35 (note that the STRESS values therein are all upscaled by a factor of 100).

Again, it is noted that the Euclidization task only focuses on minimizing the isometric disagreement between the metric defined by a given reference color difference formula and the Euclidean metric on the corresponding embedded color space, which constitutes a theoretical function approximation problem detached from experimental visual data. Accordingly, evaluating the resulting Euclidean color space on experimental visual data as above is only meant to provide additional information on the performance variation of the considered color difference system prior to and after Euclidization, which cannot replace the rigorous efficacy assessment of the proposed Euclidization approach presented in Section 6, nor should this be confused with the quality assessment of existing color difference formulas.

7.3 | Outlook on color space modeling with visual data

Considering the promising Euclidization results of the proposed neural network color space transformations, it is natural to discuss the feasibility of carrying over the present machine learning approach to the task of modeling a perceptually uniform color space by means of data from visual experiments, similarly to the development of the various performance-enhancing color difference formulas based on the CIELAB color space. First of all, it should be emphasized that the present approach to color space Euclidization relies on unsupervised learning with randomly generated color difference vectors as training examples, which in

particular excludes the issue of overfitting. In contrast, when employed for approximating a perceptually uniform color space according to experimental visual data, the neural network would be trained in a supervised manner and the performance of the resulting color model would depend on the extent and distribution of the underlying training data. Ideally, the labeled data in such a context, that is, example color pairs labeled with their perceived visual differences (ΔV_i), should be uniformly distributed across the original color space so as to prevent local distortions in the resulting color model, and the amount of them should be proportional to the size of the neural network so as to prevent overfitting. Under this consideration, existing experimental visual datasets, such as those employed for deriving the DE2000 color difference formula (cf. BFD-P,³¹ Leeds,³² RIT-DuPont,³³ and Witt³⁴), may need further enhancement to support a highly accurate neural network approximator for a perceptually uniform color space. As a solution, more extensive training data may be obtained, for instance, by conducting supplementary visual experiments with particular attention paid to the above aspects, or by augmenting existing experimental visual data with artificial training data, for example, through interpolation techniques, noise injection,³⁶ or GANs.²⁵ More study on this topic is left to future work.

7.4 | Some notes regarding further related works

In view of the plethora of vision-related machine learning efforts, it is reasonable to address how the present article relates to those works. The subsequent discussion includes a small selection of references on adjacent issues, which serve as representatives for drawing comparison with and expanding perspective.

From an interdisciplinary point of view, machine learning methodology and color-related issues based on human visual perception are both strongly represented in

image processing and computer vision. A typical class of instances is represented by deep learning approaches to perceptual-quality-based image compression, for example, Reference 37, and tone mapping of high dynamic range (HDR) scenes, for example, References 38,39, where multiscale decompositions based on Laplacian/Gaussian pyramid⁴⁰ or/and bilateral filtering⁴¹ are encapsulated in various forms in the early layers of deep convolutional neural networks (CNNs) for encoding visual information within an image (also compare Reference 42 for a typical deep CNN architecture (U-net) for hierarchical multiscale encoding and decoding). It is worth noting that those multiscale decompositions on the input side of CNNs resemble the early stages of human visual response in the sense that they reflect the bandpass/lowpass nature of human contrast sensitivity functions (CSFs) in the spatial frequency domain. Recently, Akbarinia et al.⁴³ investigated CSFs within pretrained task-dependent deep neural networks (mostly CNNs) through feature extraction and transfer learning, which revealed that human-like CSFs can be recognized with various degrees of similarity at different layers within a neural network and across different neural networks, depending on the considered image processing task.

The unifying feature of the works mentioned so far is that they are all in one way or another concerned with image processing by means of deep neural networks, that is, a two-dimensional image signal is processed through a neural network, for which building blocks that excel in analyzing information within a spatial context are employed, such as by exploiting the translation invariance of two-dimensional convolutional layers and pooling layers. In the present work, on the other hand, the color information is present in its pure form as a color coordinate triple, and accordingly, the neural network employed for processing this information falls within the category of fully connected architectures and need not even be deep. In particular, for tackling the present Euclidization problem, there are no spatial patterns to be analyzed, and deep CNNs that are so ubiquitously being exploited in other vision-related works are not well suited in this context. Actually, the same applies to modeling perceptual uniformity in general: in CIE colorimetry, color difference modeling is based on psychophysical experiments where the difference between two plain color patches on a uniform background is identified at a time (cf. e.g., References 22,31,33); in particular, no complex patterns are considered and deep CNNs that may aid in modeling spatial visual response would not be useful in deriving a perceptually uniform color space from existing experimental visual data (also recall discussion in Section 7.3).

However, it may be beneficial in general to explore the impact of taking a perceptually uniform color space as working domain on the performance of the aforementioned machine learning approaches in perception-based image processing contexts. For instance, the neural network color space transformations derived in the present work may be employed as color space pre- or post-transformations at the input or output layers of existing image processing neural network architectures. The research in Reference 44 on color conversion in variational autoencoders suggests that this type of pre- or post-coding may indeed affect the performance of the original neural networks.

8 | CONCLUSION

In this work, a general methodology making use of machine learning techniques is proposed for the issue of color space Euclidization. A parsimonious neural network architecture is employed herein and training is conducted in an unsupervised manner based on random sampling. The impact of the key disagreement measure on the color space embedding is elucidated through a distinction of cases where either the relative or the absolute isometric disagreement forms the basis of the objective function. Evaluation on well-established color difference formulas demonstrates that the proposed approach is able to deliver powerful approximators for the desired color space transformations and outperforms previous contributions in respect of commonly used average disagreement measures.

AUTHOR CONTRIBUTIONS

Lia Ahrens developed and implemented the machine learning algorithm. Julian Ahrens provided the differential geometric background information and verified and visualized the evaluation results. Hans D. Schotten contributed further to the technical background and ensured the availability of the necessary equipment.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Dr. Philipp Urban for providing the program performing the transformations in Reference 12. Furthermore, the authors would like to express their gratitude towards Senior Editor Dr. Michael Murdoch and the three anonymous reviewers for their valuable comments.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The trained neural network color space transformations and the test data generation procedure as well as scripts

for verifying the tables and figures in this article are available in the following Git repository: <https://github.com/julian-ahrens-dfki/mlacse>. Therein, only the six primary scenarios introduced in Section 4 are included.

ORCID

Lia Ahrens  <https://orcid.org/0000-0002-9382-4732>

ENDNOTES

- ¹ The probability distribution of the test color difference vector $P^{\Delta x}$ and the corresponding sampling procedure for the color difference vector Δx will be specified for each of the considered example scenarios in Section 4.
- ² In this context, the isometric disagreement loss term takes the role of a sparsity penalty. The joint neural network can therefore be regarded as a variant of a sparse autoencoder (cf. Reference 19, sec. 14.2.1).
- ³ Recall that in References 12,17, the reason for separating the L-coordinate from the ab-plane of CIELAB is that the lightness difference is calculated independently from chroma and hue in all considered color difference formulas, DE2000, CMC, and CIE94. Of course, it is possible to employ a neural network architecture without this type of channel separation, which indeed exhibited no significant performance difference during experimentation. However, since the absence of cross-contamination between lightness and chroma (as is the case with CIELAB) is a desirable feature in practical applications such as image processing tasks, a separation of the L-channel from channels associated with the ab-plane is also imposed in the present neural network transformations.
- ⁴ Here, the case of embedding the lightness coordinate according to CIE94 is an exception, as the corresponding optimal transformations are by definition the trivial identity mapping and thereby do not require any approximation.
- ⁵ In case of CIE94, the transformations associated with the L-coordinate in the upper half of Figure 6 are replaced by the trivial identity mappings (recall Footnote 4).
- ⁶ In the present approach, since unlimited training examples can be generated through random sampling, overfitting may not occur, so that the training result may be improved indefinitely. However, the inverse training typically progresses very slowly; in order to limit training time, a baseline value that is accurate enough in practice is taken here as stop criterion for the inverse approximation.
- ⁷ Recall that for all three considered example color difference formulas, the L-coordinate is processed independently from the ab-plane of CIELAB (cf. Section 5.1, in particular Figure 6 and Footnote 3). Therefore, the transformation of the ab-plane is invariant under changes of L^* , which applies to all upcoming figures, Figures 7–16. (In those figures, the only part that depends on L^* is the auxiliary sRGB gamut plotted on the background.)
- ⁸ In order to maximize the similarity of the transformed ab-plane with the familiar ab-plane of the CIELAB color space, in these plots, the transformed ab-plane is first translated so that the white point again resides at coordinates (0,0) and then rotated and, if necessary, mirrored so that the average squared difference between the CIELAB color coordinates and the coordinates in

the transformed ab-plane is minimized. Note that all performance metrics considered in the evaluation are based on the Euclidean distance and thus invariant under translations, rotations, and reflections of the transformed color space.

- ⁹ In order to avoid artifacts caused by sampling the color difference vectors randomly, a deterministic procedure is used herein. This procedure consists in generating a regular grid within a cube around the color center with an edge length of $2 \cdot 5 = 10$ CIELAB-units and then selecting from this grid only the points that have a distance of at most 5 CIELAB-units from the center.
- ¹⁰ Here and in the sequel, Figures 11–13 are generated analogously to Figures 7–9, respectively; for more details on the generation procedure for those figures, the reader is referred to the last paragraph of Section 6.1.1, in particular Footnotes 7–9.
- ¹¹ Again, Figures 14–16 are generated analogously to Figures 7–9, respectively; for more details on the generation procedure for those figures, the reader is referred to the last paragraph of Section 6.1.1, in particular Footnotes 7–9.
- ¹² In this evaluation, the four separate datasets are preferred over the weighted combined dataset, COM-Weighted,¹⁰ for excluding ambiguity in terms of the degrees of freedom required in the subsequent F -test. A detailed discussion on this can be found in Reference 35.

REFERENCES

- [1] MacAdam DL. Visual sensitivities to color differences in daylight. *J Opt Soc Am.* 1942;32:247-274.
- [2] Silberstein L. Investigations on the intrinsic properties of the color domain. II. *J Opt Soc Am.* 1943;33:1-10.
- [3] CIE. Progress report of CIE TC – 1.3 colorimetry. *Compte Rendu 18e Session, ser. Vol 36.* CIE Publication; 1975:161-172.
- [4] Luo MR, Cui G, Li C. Uniform colour spaces based on CIE-CAM02 colour appearance model. *Color Res Appl.* 2006;31:320-330.
- [5] Li C-J, Li Z-Q, Wang Z-F, et al. Comprehensive color solutions: CAM16, CAT16 and CAM16-UCS. *Color Res Appl.* 2017;42:703-718.
- [6] Clarke FJJ, McDonald R, Rigg B. Modification to the JPC79 colour-difference formula. *J Soc Dyers Col.* 1984;100:128-132.
- [7] Luo MR, Rigg B. BFD(l:c) colour-difference formula. Part I – development of the formula. *J Soc Dyers Col.* 1987;103:86-94.
- [8] CIE 116:1995. *Industrial Colour-Difference Evaluation. Tech. Rep.* CIE Central Bureau; 1995.
- [9] Cui G, Luo MR, Rigg B, Roesler G, Witt K. Uniform colour spaces based on the DIN99 colour-difference formula. *Color Res Appl.* 2002;27:282-290.
- [10] Luo MR, Cui G, Rigg B. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Res Appl.* 2001;26:340-350.
- [11] ISO/CIE 11664-6:2014. *Colorimetry – Part 6: CIEDE2000 Colour-Difference Formula. Tech. Rep.* CIE Central Bureau; 2014.
- [12] Urban P, Rosen MR, Berns RS, Schleicher D. Embedding non-Euclidean color spaces into Euclidean color spaces with minimal isometric disagreement. *J Opt Soc Am.* 2007;24:1516-1528.

- [13] Lee JM. Introduction to Riemannian manifolds. *Ser. Graduate Texts in Mathematics*. Vol 176. 2nd ed. Springer; 2018.
- [14] Helmholtz HV. Versuch, das psychophysische Gesetz auf die Farbenunterschiede trichromatischer Augen anzuwenden. *Z Psychol*. 1892;3:1-20.
- [15] Thomsen K. A euclidean color space in high agreement with the CIE94 color difference formula. *Color Res Appl*. 2000;25:64-65.
- [16] Völz HG. Euclidization of the first quadrant of the CIEDE2000 color difference system for the calculation of large color differences. *Color Res Appl*. 2006;31:5-12.
- [17] Ridolfi L, Gattass M, Lopes H. Investigating Euclidean mappings for CIEDE2000 color difference formula. In: *IS&T/SID 18th Color Imaging Conference (CIC)*; 2010:327-333.
- [18] Lissner I, Urban P. Toward a unified color space for perception-based image processing. *IEEE Trans Image Process*. 2012;21:1153-1168.
- [19] Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016. <http://www.deeplearningbook.org>
- [20] García PA, Huertas R, Melgosa M, Cui G. Measurement of the relationship between perceived and computed color differences. *J Opt Soc Am*. 2007;24:1823-1829.
- [21] Bergh J, Löfström J. Interpolation spaces, an introduction. *Ser. Grundlehren der Mathematischen Wissenschaften*. Vol 223. 1st ed. Springer-Verlag; 1976.
- [22] Guan S-S, Luo MR. Investigation of parametric effects using small colour differences. *Color Res Appl*. 1999;24:331-343.
- [23] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2016:770-778.
- [24] Kingma DP, Ba JL. Adam: a method for stochastic optimization. In: *International Conference on Learning Representations (ICLR)*; 2015.
- [25] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*. Vol 27. NIPS; 2014:2672-2680.
- [26] Muller ME. A note on a method for generating points uniformly on n-dimensional spheres. *Commun ACM*. 1959;2:19-20.
- [27] Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*. Vol 32. NeurIPS; 2019:8024-8035.
- [28] Sharma G, Wu W, Dalal EN. The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. *Color Res Appl*. 2005;30:21-30.
- [29] Pointer MR, Attridge GG. Some aspects of the visual scaling of large colour differences. *Color Res Appl*. 1997;22:298-307.
- [30] Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell*. 2016;38(2):295-307.
- [31] Luo MR, Rigg B. Chromaticity-discrimination ellipses for surface colours. *Color Res Appl*. 1986;11:25-42.
- [32] Kim D-H, Nobbs JH. New weighting functions for the weighted CIELAB colour difference formulae. *AIC Colour 97*. Vol 1. AIC; 1997:446-449.
- [33] Berns RS, Alman DH, Reniff L, Snyder GD, Balonon-Rosen MR. Visual determination of suprathreshold color-difference tolerances using probit analysis. *Color Res Appl*. 1991;16: 297-316.
- [34] Witt K. Geometric relations between scales of small colour differences. *Color Res Appl*. 1999;24:78-92.
- [35] Melgosa M, Huertas R, Berns RS. Performance of recent advanced color-difference formulas using the standardized residual sum of squares index. *J Opt Soc Am*. 2008;25(7):1828-1834.
- [36] Holmström L, Koistinen P. Using additive noise in back-propagation training. *IEEE Trans Neural Netw*. 1992;3:24-38.
- [37] Wang J, Duan Y, Tao X, Xu M, Lu J. Semantic perceptual image compression with a Laplacian pyramid of convolutional networks. *IEEE Trans Image Process*. 2021;30:4225-4237.
- [38] Rana A, Singh P, Valenzise G, Dufaux F, Komodakis N, Smolic A. Deep tone mapping operator for high dynamic range images. *IEEE Trans Image Process*. 2020;29:1285-1298.
- [39] Guo C, Jiang X. Deep tone-mapping operator using image quality assessment inspired semi-supervised learning. *IEEE Access*. 2021;9:73873-73889.
- [40] Burt PJ, Adelson EH. The Laplacian pyramid as a compact image code. *IEEE Trans Commun*. 1983;31:532-540.
- [41] Durand F, Dorsey J. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans Graph*. 2002;21:257-266.
- [42] Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: *Int Conf Med Image Comput Comput-Assist Intervent*; 2015:234-241.
- [43] Akbarinia A, Morgenstern Y, Gegenfurtner KR. Contrast sensitivity function in deep networks. *Neural Netw*. 2023;164:228-244.
- [44] Akbarinia A, Gil-Rodriguez R. Color conversion in deep auto-encoders. *J Percept Imaging*. 2021;4:20401-1-20401-10.

AUTHOR BIOGRAPHIES

Lia Ahrens received her Dr. rer. nat. degree in Mathematics from Kiel University (CAU), Germany, in 2015, specializing in stochastics and financial mathematics. She is currently working as a Senior Researcher with the Department for Intelligent Networks, German Research Center for Artificial Intelligence, under the direction of Prof. Hans D. Schotten. Her research interests include machine learning, stochastic filtering, and digital signal processing.

Julian Ahrens received his master's degree in mathematics from Kiel University (CAU), Germany, in 2016, specializing in non-commutative harmonic analysis. He is currently working as a Researcher with the Department for Intelligent Networks, German Research Center for Artificial Intelligence, under the direction of Prof. Hans D. Schotten. His research interests include digital signal processing, machine learning, high-performance computing, and harmonic and functional analysis.

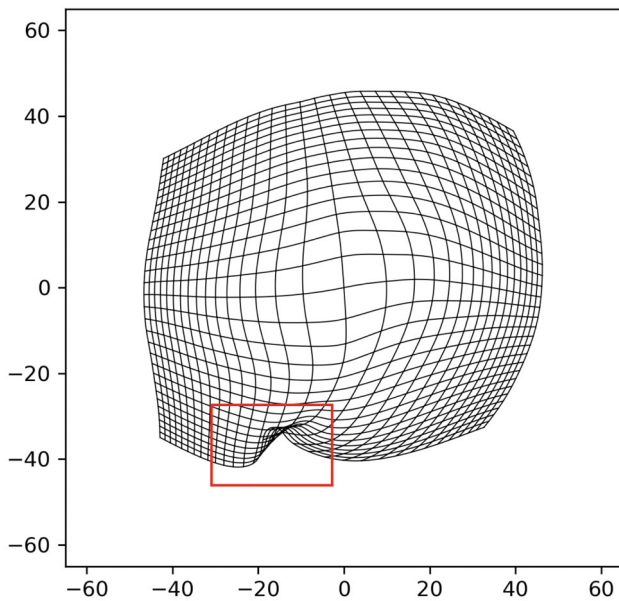
Prof. Dr.-Ing. **Hans D. Schotten** is Scientific Director of the Intelligent Networks Research Department at the German Research Center for Artificial Intelligence (DFKI) and holds the Chair for Wireless Communication and Navigation at the Rheinland-Pfälzische

Technische Universität Kaiserslautern-Landau (RPTU), Kaiserslautern, Germany. He is chairman of the Informationstechnische Gesellschaft (ITG) and member of the board of the “Verband der Elektrotechnik” (VDE, Association of Electrical Engineering). His research interests include mobile communication technologies and security technologies.

How to cite this article: Ahrens L, Ahrens J, Schotten HD. A machine learning approach to color space Euclidization. *Color Res Appl.* 2024; 49(1):4-33. doi:10.1002/col.22897

APPENDIX A

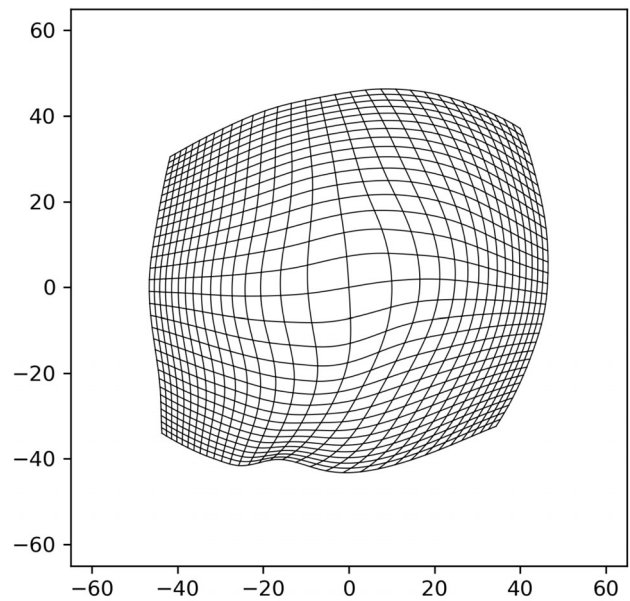
The subsequent (counter-)example aims to demonstrate that, depending on the color difference formula and the composition of the objective function, local noninjectivities in the neural network color space transformation may indeed be a valid concern, which in turn can be resolved by imposing the proposed regularization measures during training.



(A) Transformation trained without regularisation measures

As a basis, consider the following example scenario introduced in Section 4: $\Delta E := \text{DE2000}$, $d := 3$, $(r_{\min}, r_{\max}] := (0, 5]$, and the mean squared absolute isotropic disagreement is to be minimized through training. Now, as a modification of the above setting, alter the probability distribution of the test color difference vector, $P^{\Delta \mathbf{x}}$, by defining it as the probability distribution of the random vector $\Delta \mathbf{x} := \mathbf{r} \odot \mathbf{v}$ where \mathbf{v} again follows the uniform probability distribution on the unit sphere $S^2 \subseteq \mathbb{R}^3$, \odot denotes the Hadamard product, and $\mathbf{r} := (r_0, r_1, r_2)$ is a random vector with independent components, r_0, r_1, r_2 , each following the uniform probability distribution on the interval $(r_{\min}, r_{\max}]$. Accordingly, for the training data generation, adapt Step (ii) in Section 4 to the modified setting by replacing the random numbers $r^{t,n,k}$ therein with random vectors $\mathbf{r}^{t,n,k}$ and recording $\Delta \mathbf{x}^{t,n,k} := \mathbf{r}^{t,n,k} \odot \mathbf{v}^{t,n,k}$. In fact, the modification from $\Delta \mathbf{x} := r \cdot \mathbf{v}$ into $\Delta \mathbf{x} := \mathbf{r} \odot \mathbf{v}$ alters the probability distribution of both the magnitude and the direction of the color difference vector $\Delta \mathbf{x}$; the above distribution of the vector $\mathbf{r} = (r_0, r_1, r_2)$ favors the absolute distances midway in between $(r_{\min}, r_{\max}]$ and the directions near the coordinate axes.

In the above modified setting, iterating Subroutine R1 in Section 3.5 alone leads to a noninjective color space transformation, as displayed in Figure A1A. Notice how the image of the ab-plane under the embedding transformation folds in



(B) Transformation trained with regularisation measures

FIGURE A1 Images of a rectangular grid in the ab-plane of CIELAB under neural network color space transformations trained according to the objective function given in the modified setting associated with the DE2000 color difference formula, with proposed regularization measures being omitted or included.

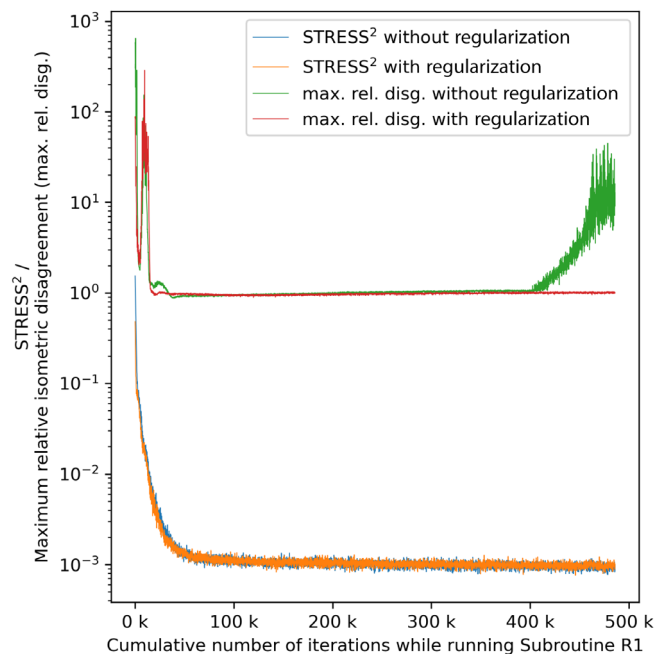


FIGURE A2 Course of forward loss (i.e., STRESS²) and maximum relative isometric disagreement while training an embedding neural network with respect to the objective function given in the modified setting associated with the DE2000 color difference formula, with versus without regularization measures.

on itself (region marked with red rectangle). In contrast, when including the proposed regularization measures in the training routine as presented in Algorithm 1, with hyperparameters $[\#R1(i) : \#R2(i)] := [2000 : 8000(1 + \lfloor i/5 \rfloor)]$ and $(\gamma := \tilde{\gamma} := 10^{-5}, \beta := 10^{-8})$, the resulting neural network color space transformation does not suffer from any local noninvertibility issues, with the corresponding ab-plane transformation visualized in Figure A1B. Figure A2 provides a comparison of the respective training course while including or omitting the proposed regularization measures. On the one hand, in the lower half of Figure A2, the two courses of training loss hardly differ from one another, which indicates that imposing the regularization term in (13) with a small factor β does not have a negative impact on the performance of the embedding transformation. On the other hand, as illustrated in the upper half of Figure A2, while the maximum relative isometric disagreement begins to explode after running Subroutine R1 for about 400k iterations continuously, this metric is kept low at a level around 1 through the alternation with Subroutine R2 in between the training iterations, which demonstrates the damping effect of the regularization measures on the local approximation error of the neural network color space transformation.