Color-Name Aware Optimization to Enhance the Perception of Transparent Overlapped Charts

Kecheng Lu, Lihang Zhu, Yunhai Wang*, Qiong Zeng, Weitao Song, and Khairi Reda

Abstract—**Transparency is commonly utilized in visualizations to overlay color-coded histograms or sets, thereby facilitating the visual comparison of categorical data. However, these charts often suffer from significant overlap between objects, resulting in substantial color interactions. Existing color blending models struggle in these scenarios, frequently leading to ambiguous color mappings and the introduction of false colors. To address these challenges, we propose an automated approach for generating optimal color encodings to enhance the perception of translucent charts. Our method harnesses color nameability to maximize the association between composite colors and their respective class labels. We introduce a color-name aware (CNA) optimization framework that generates maximally coherent color assignments and transparency settings while ensuring perceptual discriminability for all segments in the visualization. We demonstrate the effectiveness of our technique through crowdsourced experiments with composite histograms, showing how our technique can significantly outperform both standard and visualization-specific color blending models. Furthermore, we illustrate how our approach can be generalized to other visualizations, including parallel coordinates and Venn diagrams. We provide an open-source implementation of our technique as a web-based tool.**

Index Terms—**Color perception, Transparency, Overlapping charts, Simulated annealing.**

I. INTRODUCTION

OVERLAPPING, transparent visualizations are commonly used to show various data representations, from histograms to Venn diagrams to multi-variate parallel coordinates [41]. These charts create strong cues of visual superposition [18], facilitating effective comparison of distributions and sets across categorical variables. Typically, each category is depicted in a distinct color, resulting in multiple color-coded objects that are then composited using alpha blending techniques [46]. Despite their conceptual simplicity, crafting perceptually accurate translucent visualizations is challenging, requiring careful assignment of colors and opacity levels. Transparency rendering, in particular, can lead to complex interactions between colors, resulting in the introduction of false categories, or to a reduction in the coherence of the composite segments to their original class color. Moreover, resolving composite colors becomes more challenging when the original class colors are taken from hues that lie further apart in the color wheel [16], particularly when blending results do not align with recognizable or expected hues. For instance, consider the visualizations in Figure 1-b, where the blended areas suggest the presence of a new, non-existent feature (a purple 'histogram', in this case). Such artifacts can complicate the interpretation of the visualization.

To address these issues, prior work suggests the addition of visual cues to help disambiguate the visualization. For example, Wilke [66] recommends adding a kernel density estimate to each histogram. However, this approach can result in visual clutter, with multiple marks and mark-types overlapping in a potentially small space. An alternative line of research has focused on developing enhanced color-blending models that aim to reduce transparency-induced ambiguities. For instance, the *local color blending* model [61] works by desaturating 'background' colors before blending to preserve depth information (Figure 1-c). Similarly, the *hue-preserving color blending* model [8], [29] works by reducing non-dominant hues (Figure 1-d). Both of these techniques work by adjusting the blending mechanisms for a predetermined set of colors, yet they do not offer a means for users to select or generate an appropriate color palette that is resilient to blending effects, leading to suboptimal visualization. For example, a hue-preserving composite with a standard color palette (e.g., Tableau-10) frequently results in neutral, low-saturation grays that are difficult to associate with their original class (see Figure 1-d). Additionally, these techniques rely on non-standard color blending models, often requiring the implementation of custom pixel-level compositing pipelines, which may limit their applicability. Consequently, there is a need for colorization techniques that are tailored specifically for transparency considerations, yet compatible with standard graphical plotting platforms.

Inspired by recent automatic colorization methods [34], [35], we hypothesize that the perception of overlapping visualizations can be significantly improved by *directly* optimizing color assignment, opacity, and rendering order. Moreover, we hypothesize that such optimization can be achieved while utilizing a standard alpha blending model, which is widely supported in graphics and visualization toolkits. We present a novel color generation and optimization technique tailored to address the challenge of interpreting semi-transparent histograms and other categorical visualizations with overlapping elements. Our technique is grounded in the understanding that associating colors with their respective categories depends

* Yunhai Wang is the corresponding author.

Kecheng Lu & Yunhai Wang are with the School of Information, Renmin University of China, China. E-mail: lukecheng0407@gmail.com, wang.yh@ruc.edu.cn

Lihang Zhu & Qiong Zeng are with the School of Computer Science and Technology, Shandong University, China. E-mail: {leon20192050, qiong.zn}@gmail.com

Weitao Song is with the School of Optics and Photonics, Zhengzhou Research Institute, Beijing Institute of Technology, China. E-mail: swt@bit.edu.cn

Khairi Reda is with the Luddy School of Informatics & Computing, Indiana University Indianapolis, USA. E-mail: redak@iu.edu

Input palette with pre-determined color assignment, rendering order and constant opacity

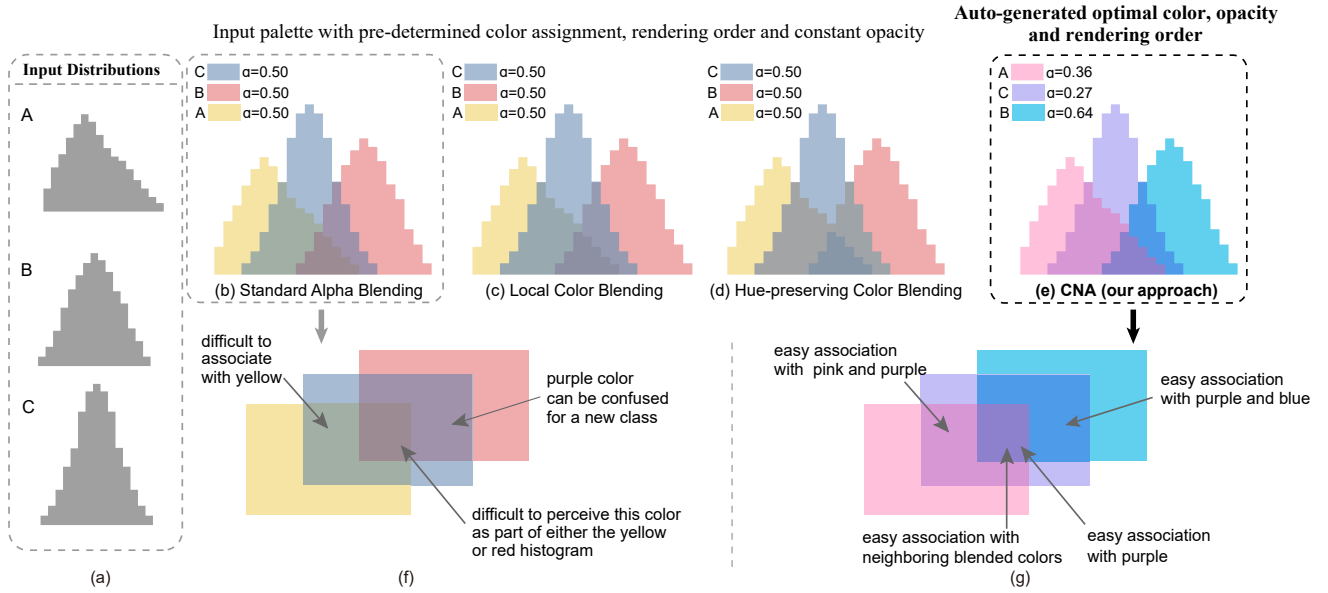Auto-generated optimal color, opacity and rendering order

Fig. 1. Results of applying our approach (right-most) to optimize three overlapping histograms. (a) shows the input component distributions; (b, c, d) illustrates the results of applying three comparable benchmarks: standard alpha blending, local color blending [61], and hue-preserving color blending [8] using base colors from the Tableau-10 palette and a uniform opacity of $\alpha = 0.5$. By comparison, our approach (e), which embodies a color-name aware optimization, auto-generates optimal color, transparency and rendering order settings, ensuring discriminability for all segments while improving whole-from-parts perception. Our technique can also optimize other overlapped visualizations, including Venn diagrams (f vs. g).

not only on color appearance but also on semantics, such as color names [65]. Leveraging this insight, we generate color assignments for overlapping elements by maximizing name similarity between parts that make up the distribution , thus facilitating effective whole-from-parts perception. Our optimization framework takes into account various data characteristics, including the arrangement of distributions and the extent of overlap, ensuring coherence within color classes while maintaining visibility for smaller segments. To account for the potential complexity of color interactions, we employ a custom simulated annealing algorithm [1] to explore the solution space and pinpoint an optimal configuration. The resulting visualizations enable accurate inference of distribution shapes while still maintaining discriminability of smaller distributional features, even when they overlap (Figure 1-e & g).

We conducted two crowdsourced studies to evaluate our approach[1]. In the first study, we compared our method against custom color blending models intended for enhancing translucent visualizations. We find that our approach yields the best performance for histogram analysis among the benchmarks. In a second study, we test whether our optimization is still effective when histograms are augmented with kernel-density estimates, finding that our technique can still improve interpretation over the baseline. We further demonstrate the generalizability of our approach beyond histograms with case studies, showing its utility for parallel coordinate plots and set visualizations. To help disseminate this method, we developed a web-based implementation of our technique as an interactive

color-design tool. To summarize, our main contributions are:

- A novel Color-Name Aware (CNA) optimization approach that automatically generates appropriate color palettes and assigns opacity values and rendering order for overlapping, semi-transparent histograms.

- Empirical validation of our approach. We show that our method allows for accurate analysis of distributional features, yielding better judgments than existing approaches.

- An extension of our method to other multi-class visualizations with overlap, including parallel coordinates and Venn diagrams.

- Finally, we contribute an implementation of our method as an open-source tool, available at https://anon-link. github.io/transparency/.

II. RELATED WORK

We survey previous work on utilizing transparency for visualization. We then discuss color optimization techniques broadly and address the perception of histograms and distribution visualizations.

A. Optimizing Semi-transparent Visualizations

Transparency plays an important role in computer graphics [13] and visualization [64] and is often exploited to reduce occlusion in 3D objects. We limit our discussion to techniques related to information visualization, and to color blending models, opacity optimizations, as well as the perception of 2D transparent visualizations.

Color Blending Models. The most common approach for simulating partial transparency is the standard alpha blending

---

[1]Experimental data and analysis code are provided as supplemental materials, and can be accessed at https://osf.io/xevk9/?view_only= 1b79aeefec774209ad60f1a5b0cceda2

model using the Porter-Duff operator [46]. However, this method often produces false colors (e.g., blending red and blue to produce purple), which can confuse the viewer. To address this issue, several alternative blending models have been proposed. Wang et al. [61] found that blending opponent colors produces neutral (e.g., greys) as opposed to categorically distinct colors. They accordingly propose to alleviate false color using a local blending approach that reduces the saturation of the background color. This results in blended color appearing closer to the front (e.g., see Figure 1-c), thus improving depth perception. Although useful in 3D settings, this technique can impede the perception of the background objects. Furthermore, local blending cannot entirely eliminate false colors. Chuang et al. [8] developed a hue-preserving blending model to address the issue of false colors. This approach works by modifying the hue component of the non-dominant color such that it is the opponent of the dominant hue. The results thus preserve the hue of the dominant color. However, as shown in Figure 1-d, this method often results in the complete loss of hue (except for the dominant color), which is a serious limitation for visualizations that rely extensively on color-encoding to communicate categories. Instead of heuristics-based optimization, Zhang et al. [29] propose a data-driven color blending model based on a lab study that specifically attempts to avoid false colors and preserve depth ordering, and demonstrate the effectiveness of their method with a use case based on interpreting parallel coordinates. Unfortunately, the collected data and model have not been released and thus we cannot reproduce their results. Almost all these methods aim to generate effective visualization for better preserving depth order, whereas our goal is to help users perform distributional analyses like identifying and comparing the shapes of distributions.

In lieu of compositing colors, Urness et al. [60] propose an alternative method to convey transparency via color weaving [60], that interweaves individual colors of multiple variables to form a high-frequency texture. A user study by Hagh-Shenas et al. [23] found that color weaving is more efficient than traditional color blending at conveying the values of individual data distributions in a multivariate visualization, but the performance degrades as the number of overlapping classes increases. Luboschik et al. [37] extended this technique for representing overlapping regions in scatterplots. Although promising, the use of color weaving is uncommon in practice and could lead to poor performance for some comparison tasks. Furthermore, this technique is highly sensitive to choices of textural and statistical parameters [3].

Opacity Optimization. Specifying proper opacity values can be an effective way to reduce the occlusion. A few opacity optimization techniques have been developed for a variety of 2D/3D objects, including lines [20], [36], surfaces [21], and volume [63]. To address over-plotting in parallel coordinates, Johansson et al. [27] and Zhou et al. [69] introduce tools that assign opacity based on local data density. However, these tools still require considerable manual effort to optimize the visualization. Based on a crowdsourced study, Matejka et al. [40] proposed a user-driven opacity scaling model for scat-

terplots that automatically assigns an appropriate transparency for each data point. Micallef et al. [43] and Quadri et al. [47] extended this effort by considering additional visual variables alongside mark opacity to better support certain data analysis tasks. All of these optimization techniques do not consider color blending artifacts caused by composite colors, which is a central focus of our work.

Transparency Perception. Transparency can serve as an important 'channel' in visualization, with several models having been developed to measure and quantify the effect of this cue. For example, Metelli's episcotister model [42] is one of the earlier models to quantify the perception of transparency. The X-junction model [2] utilizes luminance to predict depth perception. These models can be used to optimize transparency configurations for volume visualizations [68]. Beck et al. [6] test the Metelli model through multiple experiments, and report that the degree of perceived transparency varied linearly with lightness, not the reflectance. Similarly, Singh et al. [57] conduct further experiments toward building a perceptually based theory of transparency that addresses the limitations of the episcotister model. Readers are referred to a recent survey [17] for a more detailed review of the translucency perception. Gama and Gonçalves [15] investigate people's ability to resolve the original colors in blended composites. Their findings indicate that participants perform poorly when resolving opponent hues that are composited together, or when identifying source colors from three (as opposed to two) colors. Notably, blending performed in the CIE LCh model space resulted in higher accuracy compared to other color spaces. Our research similarly seeks to create translucent charts that enhance the 'provenance' of composite colors for categorical visualization, which is important for distribution analysis, among other types of data.

B. Color Optimization

Once a source palette is chosen (e.g., from Color-Brewer [24]), an important step in categorical visualization design entails assigning colors from the palette to different categories. A few optimization techniques have been proposed to streamline this process, often with the goal of maximizing semantic association between colors and the concept they represent. For example, Lin et al. [33] propose semantically resonant color assignment for categorical data based on a semantic affinity score calculated from a set of representative images for each category. Setlur and Stone [55] refine this model and introduce a semantic co-occurrence measure that leverages color-name frequencies. El-Assady et al. [12] developed a pipeline for mining semantic color associations from the literature. Schloss et al. developed a framework to facilitate colormap interpretation by ensuring a level of 'semantic discriminability' for the represented categorical variables [54], [53], [44]. However, some categorical visualizations might not contain clear semantic associations. Our technique instead relies on optimizing color name consistency to provide additional cues for discriminating displays with overlapping and potentially crowded elements. The impact of color names has been emphasized in recent studies, with results suggesting an

important role for nameability in colormap interpretation [50], [49], [48]. Wang et al. [62] propose a scheme aimed at maximizing the discriminability of multi-class scatterplots. This approach considers the spatial relationships between classes along with color contrast against the background. Subsequently, Lu et al. [34] introduced Paletailor, a framework that combines palette generation and color assignment into a unified optimization process, showing its effectiveness for various types of charts. Most recently, Languenou [30] extends the technique introduced by Wang et al. [62] to deal with area-based visualizations like streamgraphs and chord diagrams. All of these techniques, however, are tailored to opaque visualizations, making them less suited for application translucent visualizations.

C. Perception of Data Distributions in Histograms

Histograms are commonly used to visualize univariate distributions, supporting several distribution analysis tasks [7]. However, understanding such charts is not easy and is influenced by different factors. Lem et al. [32] and Kaplan [28] show that students have misinterpretations in identifying frequencies of specific data values within histograms even when they have the requisite knowledge and adequate time. Correll et al. [10] found that the number of histogram bins heavily impacts the tasks of detecting missing values and outliers. Sahann et al. evaluated the influence of the bin number, finding that accuracy stabilizes around 20 bins and does not improve by adding additional bins [52].

To simultaneously compare multiple distributions, overlapping histograms are a widely used method [66], even if they frequently suffer from false colors introduced by color blending. The most prevalent way to alleviate this problem is to overlay kernel density curves to accentuate the underlying. Blumenschein et al. [7] introduced v-plots that combine mirrored bar charts and violin-style plots, which are designed to aid the comparison of distributions at different levels. However, these plots are limited to comparing two distributions only at a time. By contrast, translucent color-coded histograms can theoretically support a larger number of distributions within the same chart. In this paper, we attempt to enhance the perceptibility of overlapping histograms (and other types of visualizations employing transparency) by optimizing color and opacity settings, as well as the rendering order.

III. METHOD

Given a set of m distributions $\mathbf{X} = \{1, \cdots, m\}$ over a specific background color c_b, our goal is to find a proper color c_i, an opacity value $\alpha_{\mathbf{i}}$ and the rendering order o_i for each distribution. Our method can be combined with any color blending method. However, because we intended for our method to work with the standard Porter-Duff alpha-blending model [46], we employ the latter for all illustrations and evaluations in this paper, without loss of generality. Specifically, when blending two colors c_i and c_j with opacity values α_i and α_j, respectively, the color compositing operator

generates a blended color c' and a corresponding opacity α' using linear interpolation:

$$\alpha' = \alpha_i + \alpha_j(1 - \alpha_i)$$
$$c' = \frac{\alpha_i c_i + \alpha_j c_j(1 - \alpha_i)}{\alpha'}$$

We illustrate our method with overlapping histograms, as they represent one of the most common chart types employing transparencies. We assume that each distribution has at least one non-overlapping section, and apply the above operator to render m distributions, yielding n (where $n > m$) parts and $n - m$ overlapping regions where the resulting color is a blend of multiple colors. To convey the membership of each overlapping region to its respective parent distribution(s) and to meet the needs of categorical data visualizations [62], [34], we propose three design requirements:

(i) **DR1:** a blended color should have a strong association with at least one (and ideally all) of the classes it is a member of;

(ii) **DR2:** a color should not have a strong association with another color if they do not share membership in at least one class; and

(iii) **DR3:** all colors, including colors resulting from the blending process, should be sufficiently discriminable. Smaller segments, in particular, should have high contrast against surrounding colors.

To meet DR1 and DR2, we reinforce the association between color and category by leveraging color nameability [25], [49]. Specifically, we ensure that blended segments in the visualization retain a high name similarity with their original class color, while minimizing name similarity among segments belonging to different categories. Simultaneously, we optimize the discriminability of all color segments to meet DR3.

A. Objective Function

Based on the three design requirements, we formulate a search for an optimal color palette $\mathbf{P} = \{c_1, \cdots, c_m\}$, a corresponding opacity set $\mathbf{A} = \{\alpha_1, \cdots, \alpha_m\}$, and a rendering order $\mathbf{O} = < o_1, \cdots, o_m >$ which collectively optimize the perception of the visualization. This optimization admits the following objective function $E(\mathbf{P}, \mathbf{A}, \mathbf{O})$:

$$\begin{aligned} E(\mathbf{P}, \mathbf{A}, \mathbf{O}) = {}& \omega_1 E_{WA}(\mathbf{P}, \mathbf{A}, \mathbf{O}) \\ & - \omega_2 E_{BD}(\mathbf{P}, \mathbf{A}, \mathbf{O}) \\ & + \omega_3 E_{CS}(\mathbf{P}, \mathbf{A}, \mathbf{O}) \end{aligned} \quad (1)$$

Where the weights $\omega_i \in [0, 1]$ dictate the relative importance of the terms (all set to 1 by default). The three terms represent, respectively, Within-class Association (WA), Between-class Disassociation (BD), and Color Separability (CS). To facilitate the computation of these three terms efficiently, we pre-construct a $n \times m$ membership matrix $\mathbf{M}$, whereas $\mathbf{M}_{i,j}$ indicates whether the ith region R_i of a histogram is part of the jth class. Figure 2-c illustrates how this membership matrix relates to the input distributions with an example. Here, $\mathbf{M}_{1,2} = 0$ indicates that the region R_1 does not belong to the second histogram (i.e., class B) where $\mathbf{M}_{1,1} = 1$ indicates that R_1 is part of the first histogram (class A). The resulting colors

of the n regions are denoted by $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_m, \cdots, \mathbf{c}_n\}$, where the first m regions, $\mathbf{c}_i$ ($i = 1, \cdots, m$), are a mixture of the primary class colors defined by $\mathbf{P}$ and the background color c_b. These colors correspond to non-overlapping regions. The remaining colors, c_j ($n \geq j > m$), are composite colors that result from blending two or more colors from the palette $\mathbf{P}$. The opacity for each of the m classes is determined by $\mathbf{A} = \{a_1, \cdots, a_m\}$, and the rendering order is determined by the sequence $\mathbf{O} = <o_1, \cdots, o_m>$. Note that our optimization is applied only to $\mathbf{P}$, $\mathbf{A}$, and $\mathbf{O}$. However, the full set of colors $\mathbf{C}$ is updated at each iteration to account for the blending results with the background and between overlapping histogram segments. We now describe each of the three optimization terms in detail.
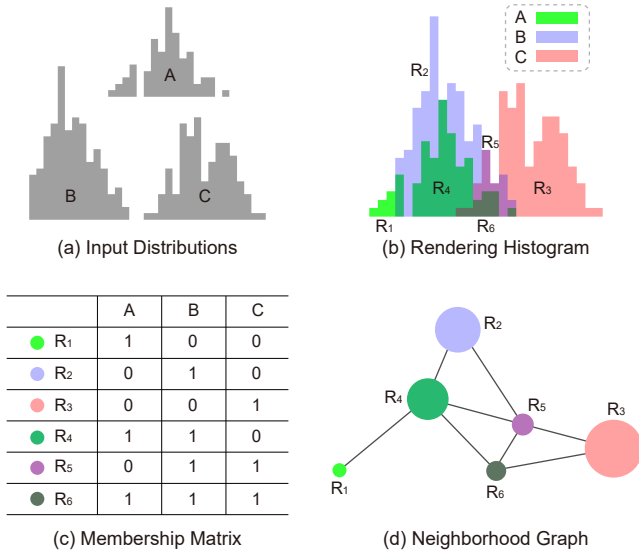

Fig. 2. Preprocessing of the objective function. The input distributions are shown in (a). Rendering these three histograms (b) results in six regions $\{R_1, ..., R_6\}$, with some regions representing an intersection of two or more histograms. The binary membership matrix (c) lists the membership for each region, with $\mathbf{M}_{i,j}$ indicating whether region i belongs to class (i.e., class) j. For example, the region R_4 belongs to the 1st class and 2nd class, which are classes A and B. The neighborhood graph (d) indicates region adjacency, with node size corresponding to the size of the region.

Within-Class Association. For each overlapping region, we define its degree of association with its 'parent' classes based on color name similarity (recall that a region may belong to more than one class). We then maximize the name similarity between the blended color and its parent colors (**DR1**) in the objective function Equation 1. Accordingly, we can define within-class association degree as:

$$E_{WA}(\mathbf{P}, \mathbf{A}, \mathbf{O}) = \frac{\sum_{i \leq m < j \leq n} \Psi(i)^{\frac{1}{2}} W_{i,j} S(\mathbf{c}_i, \mathbf{c}_j)}{\sum_{i \leq m < j \leq n} W_{i,j}} + \min_{W_{i,j} > 0} S(\mathbf{c}_i, \mathbf{c}_j) \quad (2)$$

The first term captures the *average* name similarity (S) between a histogram i's base class color and the colors of each of its constituent regions j. The second term finds the *minimum* similarity within these pairings. The rationale for including the latter is to force the optimization to maximize both the mean and the lowest within-class name similarity, ensuring

no histogram suffers from poor cohesion. Specifically, $W_{i,j}$ represents the number of classes to which both region i and region j belong, which equals the dot product of the rows i and j of membership matrix $\mathbf{M}$. For example, $W_{1,5} = 0$ indicates that the two regions R_1 and R_5, corresponding to the row vectors $(1, 0, 0)$ and $(0, 1, 1)$ in Figure 2, are not members of a shared class, thus do not contribute to the final within-class association degree. By contrast, $W_{1,4} = 1$, forcing a higher color similarity for R_1 and R_4.

$S(\mathbf{c}_i, \mathbf{c}_j)$ is the cosine-based name similarity between colors c_i and c_j proposed by Heer and Stone [25]:

$$S(\mathbf{c}_i, \mathbf{c}_j) = cos(\mathbf{T}_{\mathbf{c}_i}, \mathbf{T}_{\mathbf{c}_j}) = \frac{\mathbf{T}_{\mathbf{c}_i} \mathbf{T}_{\mathbf{c}_j}}{||\mathbf{T}_{\mathbf{c}_i}|| ||\mathbf{T}_{\mathbf{c}_j}||}, \quad (3)$$

where $\mathbf{T}$ is a color-term count matrix collected by an online study, and $\mathbf{T}_{\mathbf{c}_i}$ is the probability distribution of color names for a given color $\mathbf{c}_i$. A larger value of $S(\mathbf{c}_i, \mathbf{c}_j) \in [0, 1]$ indicates the two colors are more likely to share the same name. We incorporate a class-specific weight $\Psi(i)$ in Equation 2 based on two considerations: i) distributions that comprise a larger number of overlapping segments should be given a higher optimization priority; and ii) when the ratio of the overlapping to non-overlapping segments in one distribution is large, the corresponding class-weight is similarly increased. Accordingly, $\Psi(i)$ is defined as:

$$\Psi(i) = \frac{\sum_{j=1}^{n} \mathbf{M}_{j,i}}{\max_{i \leq m} \sum_{j=1}^{n} \mathbf{M}_{j,i}} \left(1 - \frac{RS_i}{\sum_{j=1}^{n} W_{i,j} RS_j}\right), \quad (4)$$

where $\sum_{j=1}^{n} \mathbf{M}_{j,i}$ is the number of regions that make up the ith class, computed by summing up column i in the membership matrix. The size of the i-th region, denoted by RS_i, represents the fraction of pixel space occupied by the region relative to the entire visualization footprint. Accordingly, $\sum_{j=1}^{n} W_{i,j} RS_j$ reflects the aggregate size of the ith distribution. In essence, the value of $\Psi(i)$ allows us to heuristically score the 'difficulty' of perceiving distribution i, allowing the optimization to prioritize potentially problematic features.

Between-Class Disassociation. While the within-class association term (Equation 2) ensures that a single distribution with its overlapping segments is perceived as a coherent whole, we further need to maximize color dissimilarity between different distributions so that they can be perceived as separate entities. We achieve this by defining a between-class disassociation term. Specifically, this term assigns colors with larger color-name distances to all segments that do not belong to the same distributions:

$$E_{BD}(\mathbf{P}, \mathbf{A}, \mathbf{O}) = \frac{\sum_{i < j \leq n} \delta(W_{i,j}) S(\mathbf{c}_i, \mathbf{c}_j)}{\sum_{i < j \leq n} \delta(W_{i,j})} + \max_{W_{i,j} = 0} S(\mathbf{c}_i, \mathbf{c}_j), \quad (5)$$

Where $\delta(x)$ returns 1 if $x = 0$ and 0 for all other x values. The first term computes the average color name similarity for pairs of regions that do not belong to the same class. For example, $\delta(W_{1,3}) = 1$ since R_1 and R_3 are not members of the same class, thus their similarity will be included in

the average. By contrast, $\boldsymbol{R}_5$ and $\boldsymbol{R}_6$ share two classes, with $W_{5,6} = 2$. Accordingly, $\delta(W_{5,6}) = \delta(2) = 0$ thus preventing these two regions from being penalized for having similarly sounding names. The second term finds the highest name similarity among these pairings. Combining the two terms in the objective function minimizes both the mean and the highest similarity between disjoint histograms, thereby yielding distinct colors for regions that do not belong to the same class(es). In doing so, we ensure that **DR2** is met.

Color Separability. **DR3** requires that all colors should be sufficiently discriminable. Smaller regions, in particular, should have high contrast against their surroundings so that they can be perceived sufficiently. To incorporate this notion, we define a color separability term based on a region-based neighborhood graph (see Figure 2-d for illustration). Since region size influences the perception of the color difference [59] (specifically, smaller regions require larger differences), we operationalize the latter and define our separability term as:

$$E_{CS}(\mathbf{P}, \mathbf{A}, \mathbf{O}) = \min_{\forall i \leq n, \forall R_j \in \Omega(R_i)} D(\mathbf{c}_i, \mathbf{c}_j)\big(1 + RS_i\big), \quad (6)$$

where $\Omega(R_i)$ indicates all neighborhoods of the region R_i, $D(\mathbf{c}_i, \mathbf{c}_j)$ represents the CIEDE2000 color distance [56] between two colors $\mathbf{c}_i$ and $\mathbf{c}_j$. In Equation 6, the color distance $D(\mathbf{c}_i, \mathbf{c}_j)$ is weighted by the area of the distribution $(1 + RS_i)$, as smaller areas make colors more difficult to distinguish. The additional constant ensures that the measure is not highly sensitive to very small RS_i values. This measure is computed for all neighboring pairs, with the overall performance determined by the worst separability, and thus, the minimum of all pairwise comparisons is used.

Additionally, we introduce two hard constraints for discriminability: A **just-noticeable difference** constraint ensuring that all colors are perceivable:

$$\forall i \leq n, j \leq n, i \neq j, D(\mathbf{c}_i, \mathbf{c}_j) > \eta, \quad (7)$$

where η is a just noticeable difference (JND) threshold, which is 3 by default [67]. A second, **background contrast** constraint ensures that colors sufficiently stand out:

$$\forall i \leq n, LD(\mathbf{c}_i, \mathbf{c}_b) \geq \sigma \quad (8)$$

where $LD(\mathbf{c}_i, \mathbf{c}_b)$ is the absolute luminance difference between the color $\mathbf{c}_i$ and background color $\mathbf{c}_b$ in the CIELAB space, and σ is a threshold with the default value 5 units (approximately twice the empirical JND [38]), thus ensuring that the difference is perceptible to most people.

B. Optimization Strategy

In addition to optimizing the color palette $\mathbf{P}$ and the opacity $\mathbf{A}$, we also optimize the rendering sequence $\mathbf{O}$ for the m distributions. This is crucial as the outcome of alpha blending depends on the rendering sequence, which impacts the quality of the visualization. To efficiently solve this optimization, we developed a customized simulated annealing [1] algorithm (see Algorithm 1). The algorithm starts at a high 'temperature' that is gradually 'cooled'. It begins with a random solution for each

---

Algorithm 1 Customized Simulated Annealing

1: randomly initializing color mapping $\mathbf{P}_0$ with m classes, randomly initializing the set of opacity values $\mathbf{A}_0$, randomly initializing the rendering order $\mathbf{O}_0$
2: set initial temperature T_0, cooling coefficient γ
3: set the best color mapping $\mathbf{P}_{best} = \mathbf{P}_0$, opacity sets $\mathbf{A}_{best} = \mathbf{A}_0$ and rendering order $\mathbf{O}_{best} = \mathbf{O}_0$
4: $t = 0$
5: **while** $T_t > T_{end}$ **do**
6:   set the current color mapping $\mathbf{P} = \mathbf{P}_t$, current set of opacity values $\mathbf{A} = \mathbf{A}_t$ and current rendering order $\mathbf{O} = \mathbf{O}_t$
7:   **do**
8:     $p \leftarrow random(0, 1)$
9:     **if** $p < \frac{1}{3}$ **then**
10:       randomly disturbs one color of $\mathbf{P}$
11:     **else if** $p < \frac{2}{3}$ **then**
12:       randomly changes one opacity value of $\mathbf{A}$
13:     **else**
14:       randomly exchanges two colors' rendering order of $\mathbf{O}$
15:     **end if**
16:   **while** $\exists D(\mathbf{c}_i, \mathbf{c}_j) < \eta$ **or** $\exists LD(\mathbf{c}_i, \mathbf{c}_b) < \sigma$
17:   $\Delta E = E(\mathbf{P}, \mathbf{A}, \mathbf{O}) - E(\mathbf{P}_t, \mathbf{A}_t, \mathbf{O}_t)$
18:   **if** $\Delta E > 0$ **then**
19:     $\mathbf{P}_{t+1} \leftarrow \mathbf{P}, \mathbf{A}_{t+1} \leftarrow \mathbf{A}, \mathbf{O}_{t+1} \leftarrow \mathbf{O}$
20:   **else**
21:     **if** $random(0, 1) < \exp(\Delta E / T_t)$ **then**
22:       $\mathbf{P}_{t+1} \leftarrow \mathbf{P}, \mathbf{A}_{t+1} \leftarrow \mathbf{A}, \mathbf{O}_{t+1} \leftarrow \mathbf{O}$
23:     **end if**
24:   **end if**
25:   $T_{t+1} = \gamma T_t, \quad t \leftarrow t + 1$
26: **end while**

(c_i, α_i, o_i) and updates this solution at each iteration in two major steps: i) generating a new candidate solution from the current one, and ii) comparing the new candidate to the old solution and deciding whether to accept it. In line with prior work on simulated annealing for color selection, we choose a cooling coefficient of $\gamma = 0.99$, the starting temperature is $T = 100,000$ and the ending is 0.001 [34]. We next explain the above two steps in detail.

Generating a new solution (lines 7-16): At every iteration of the optimization, we generate a random solution that is one step removed from the current solution. This is done by randomly choosing from three possible perturbation mechanisms that are equally probable (lines 9, 11, and 13): i) slightly perturbing one color in the current palette $\mathbf{P}$ with a small RGB offset (in our implementation, we set this offset to 10 units for each component), ii) adjusting the opacity value $\mathbf{A}$ for one of the classes with a small offset (0.1 by default), and iii) swapping the rendering order $\mathbf{O}$ of two randomly selected colors. The perturbation results in a new candidate solution that is then tested against the two hard constraints in Equation 7 or Equation 8. If the solution fails either of the two constraints, it is rejected, with a new candidate solution
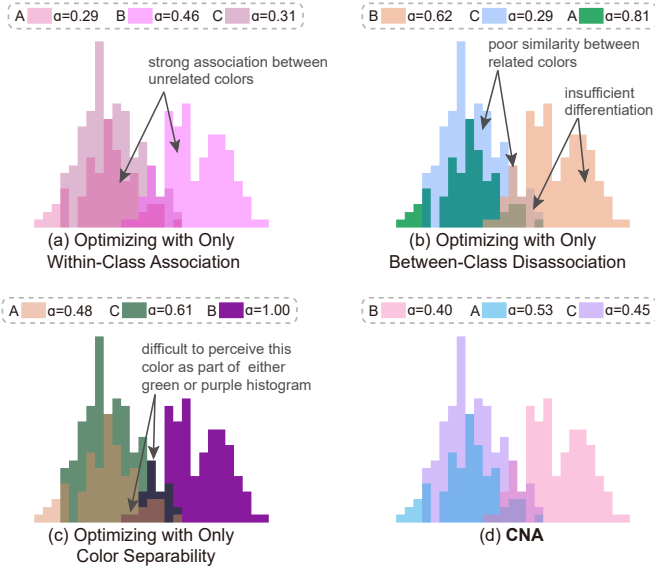
Fig. 3. The influence of different weight settings (ω_1, ω_2, ω_3) on the final colorization results, showing the contribution of each of the three objective-function terms individually. (a) illustrates the effects of optimizing within-class association only (1, 0, 0). (b) shows between-class discrimination only (0, 1, 0). (c) illustrates the effect of maximizing color separability (0, 0, 1). (d) illustrates the final result with all three terms equally weighted (1, 1, 1). The legend also depicts the optimized rendering order (from top to bottom). The input distributions are the same as those shown in Figure 2-a.

Fig. 4. Different initializations (left) and the corresponding optimization results (right). The plots of E(P,A,O) versus the number of iterations illustrate the convergence of the proposed method (middle).

generated instead. We perform color perturbation in the RGB space to avoid the challenges associated with manipulations in the CIELAB space, where even small changes can yield colors outside the RGB gamut. This requires additional constraints, such as gamut clipping or mapping, introducing unnecessary complexities into the algorithm.

Accepting a new solution probabilistically (line 17-24):
We score the new candidate solution $(\mathbf{P}_t, \mathbf{A}_t, \mathbf{O}_t)$ (line 17), accepting it if improves the score relative to the current solution. If, on the other hand, the candidate is objectively worse, we accept it with a probability of $e^{\Delta E/T_t}$, where ΔE is the difference in score between the new and the current solution and T_t is the current temperature. Higher temperatures yield a higher probability of accepting a worse solution, which reduces the likelihood of encountering local minima earlier in the optimization.

C. Parameter Analysis

The three weights (ω_1, ω_2, ω_3) in Equation 1 exert significant influence on the final results, as shown in Figure 3. When optimizing within-class association only, the overlapping regions within each distribution tend to be similar, facilitating easy grouping. Yet, establishing global membership across different distributions is still challenging. For instance, in Figure 3-a, similar colors (three pink shades) are assigned to different classes. By optimizing for between-class disassociation only (b), differentiation between classes becomes straightforward. Still, some distribution segments are rendered indistinguishable. The latter is helped by color separability, although this factor alone yields a low-quality solution (c). Conversely, by integrating all three optimization
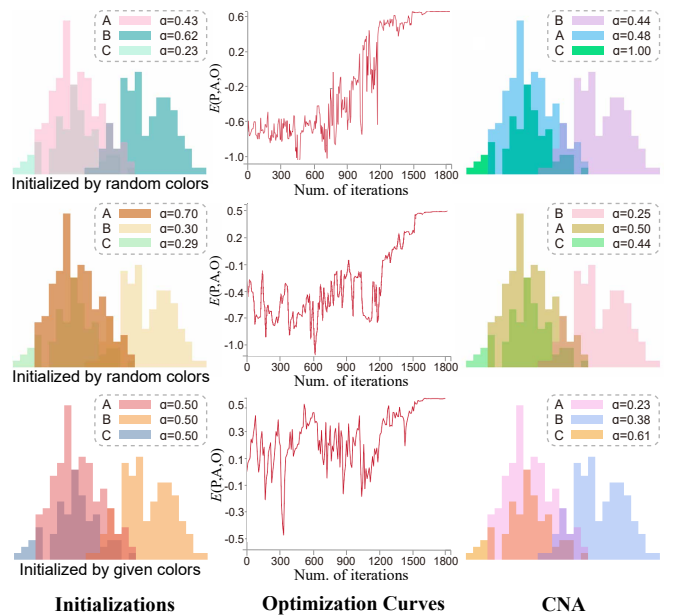
terms (d), we obtain clearer class membership along with improved separability from both neighboring colors and the background. We assume that the three factors—within-class association, between-class disassociation, and color separability—are equally important. As a result, we assign equal weights to each factor and use this combination to generate all the results presented in this paper.

Although individual results may vary between runs of simulated annealing, our approach consistently converges to a robust and effective colorization solution. As shown in Figure 4, even with different initializations, our method reliably produces satisfactory results. The convergence curves exhibit significant oscillations in the early stages, but the algorithm stabilizes at the end of the run. Notably, when the initial colors are specified rather than randomized, as illustrated in the bottom row of Figure 4, the method remains capable of identifying suitable results. However, we adopt random initialization by default as typically used in simulated annealing to encourage greater solution diversity.

IV. QUANTITATIVE EVALUATION

We present a quantitative evaluation of our optimization technique. Our primary focus is to assess the extent to which our method enhances the analysis of semi-transparent, overlapping histograms. These multi-class histograms are commonly used for the analysis of distributions (e.g., comparing the distribution of a quantitative variable across multiple class stratification). We adopt three experimental tasks (see Figure 5) and employ distributions with varying levels of smoothness and overlap. We compare our technique against three benchmarks: standard alpha blending [46], local color blending [61], and hue-preserving color blending [8]. For these benchmarks, we employ color palettes selected from either Tableau [58] or generated through Colorgorical [19], employing uniform
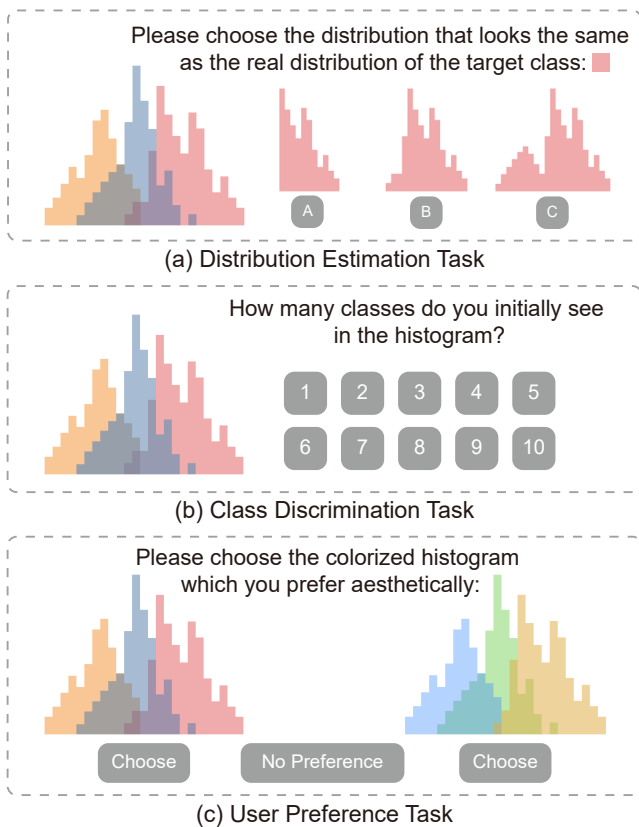
Fig. 5. Illustration for the three tasks used in our experiments: (a) *distribution estimation*, (b) *class discrimination*, and (c) *user preference*. The correct answer for the distribution estimation task is option B, while the correct number of classes in the discrimination task is three classes.

opacity and the same rendering order to simulate common usage conditions. By contrast, our optimization generates a color palette, opacity values, and rendering order tailored to the characteristics of the distributions. Additionally, because histograms are often shown accompanied by kernel density estimates [66], we test this setup in a second study §IV-B. We conducted both studies on Prolific, recruiting a total of 160 participants, ranging in age from 18 to 65 years. The minimum education level among all participants was high school, with most participants having earned a bachelor's degree.

Stimuli and Palette Generation. We created a collection of 18 multi-class stimuli (see Figure 7 in the supplementary material), each comprising overlapping histograms with two, three, or four classes, with six stimuli for each class number. These histograms displayed a diverse range of characteristics, including varying degrees of distribution smoothness and overlap. To construct each stimulus, we sampled from a Gaussian distribution with a random standard deviation ($\sigma \in [3, 5]$) and a random scaling factor ($A \in [0.8, 1.2]$). After generating the initial Gaussian, we introduced random perturbations to induce discontinuities in the distribution. The level of perturbation dictates the difficulty of resolving the histogram; histograms that are closer to an idealized smooth Gaussian are easier to perceive, as they provide sufficient visual continuity and symmetry (see Figure 6). To quantity smoothness, we employed the Kullback-Leibler (KL) divergence to gauge the dissimilarity between the perturbed distribution and its ideal Gaussian.

We categorized the smoothness into three levels: smooth, moderately smooth, and unsmooth. These levels correspond to KL values of 0, $[0.02, 0.04]$, and $[0.07, 0.1]$, respectively. We factor each smoothness level with the number of classes (two repetitions each), giving us a total of 3 *classes* $\times$ 3 *smoothness levels* $\times$ 2 *repetitions* = 18 overlapped histograms. Then for each histogram, we manually designed three response choices: One of the choices was the actual (i.e., correct) distribution. The other two options representing confounders correspond to the base histogram with overlapping segments either added or removed from the distribution.

To color-code the overlapped histograms in the benchmark conditions, we utilized two sources: a designer-crafted palette and an auto-generated palette created using Colorgorical [19]. For the designer-crafted palette, we employed the Tableau-10 categorical scheme [58], known for its quality and discriminability. From this palette, we randomly selected subsets of two, three, or four colors to assign to the classes in the stimuli. For the Colorgorical-generated palette, we utilized the low-error setting, generating a palette with eight base colors. Similarly, we randomly sampled colors from this palette, depending on the number of classes required for the stimulus. We simplified the generation process of Colorgorical for a fair comparison with Tableau, but it is worth noting that Colorgorical is designed to return an optimal palette given the number of colors queried. It is possible that randomly selecting a subset of 4 colors from an 8-color palette will result in a worse palette than querying the tool for a 4-color palette.

A. Study I: Histograms

In this experiment, we evaluate how well people perceive translucent overlapping histograms without additional visual cues. We compare our method against the three benchmarks outlined above. We utilize three tasks adapted from prior work on distribution analysis [7], [52], [62], [34], [35]:

- *Distribution Estimation*: Based on literature in histogram visualization [7], [52], we adopted a task of "describing and identifying the shape and type of a distribution." In this task, participants are asked to report the shape of the distribution for a target class, given three options (including two confounders and one correct choice, as shown in Figure 5-a). We record participants' accuracy (a binary measure of whether the response is correct) and response time.
- *Class Discrimination*: This task aims to evaluate participants' ability to accurately discern the number of distinct classes presented in the visualization, thus assessing color discriminability and the potential presence of false colors. As shown in Figure 5-b, participants are asked to count and report the number of distinct categories they perceive in the visualizations, with options ranging from 1 to 10. We similarly analyze the response time and relative error (the difference between the true number and the participant's selection). To facilitate comparisons across stimuli, we normalize the error by the number of classes in the visualization.
- *User Preference*: This task measures the aesthetic appeal of our generated colors. Participants are presented with a
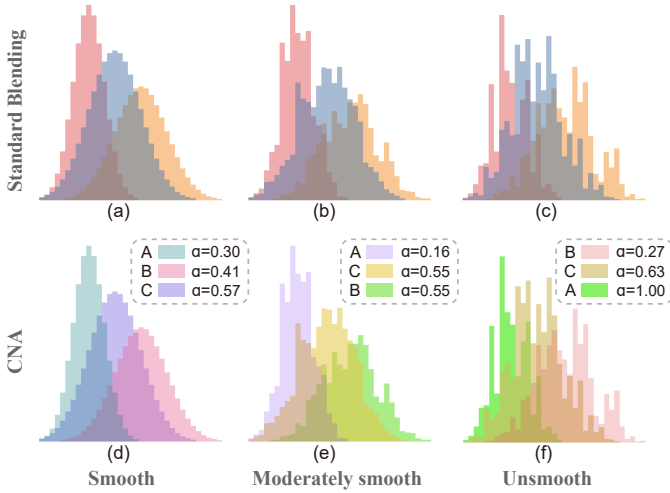
Fig. 6. The impact of distribution smoothness. (a, d) show smooth histograms representing ideal Gaussians. (b, e) are moderately smooth histograms produced by adding perturbation. (c, f) are unsmooth with an even higher level of perturbation, yielding more challenging stimuli.
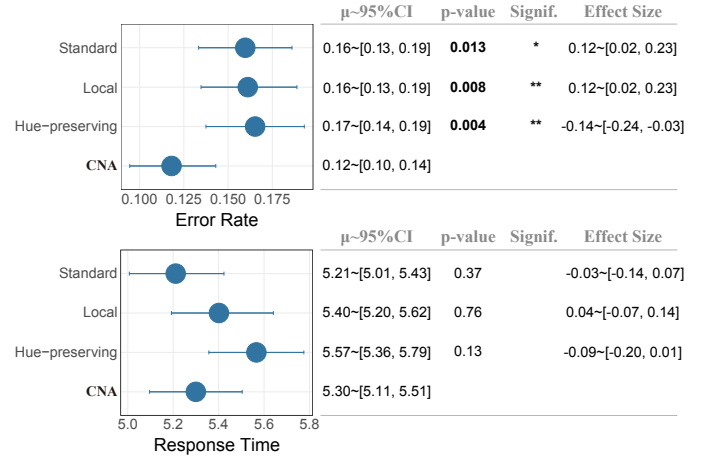

Fig. 7. Results for the *Distribution Estimation Task* from Study I, including effect sizes and significance tests for our method against the benchmarks. Error bars ($\mu \sim 95\%$CI), p-value, and significance level are calculated from the Mann-Whitney test. Effect size is calculated using Cohen's d.

pair of visualizations depicting the same histogram data but using two distinct color schemes: one derived from our optimization and the other from the benchmark sources. An example is shown in Figure 5-c. Participants are prompted to select the visualization they find more appealing.

Conditions. We conduct a comparison of our approach (referred to as *CNA*) against three benchmarks: traditional alpha blending (*Standard*), local color blending (*Local*) and hue-preserving blending (*Hue-preserving*). The latter two are visualization-specific blending approaches: the hue-preserving model is designed to minimize the appearance of false features by ensuring that the blended colors maintain a consistent hue with the original class color [8]. On the other hand, local blending seeks to reduce false colors by preserving the perceived order of color layers [61]. This is achieved by selectively desaturating 'background' objects while retaining similar hue and lightness levels. For the three benchmarks, we sampled colors from the Tableau-10 scheme or Color-gorical (see 'Palette Generation'). To assign opacity values, we followed the guidelines from Matplotlib [51], adopting a uniform opacity of 0.5. By contrast, our approach generates colors, opacity values, and rendering order algorithmically according to the optimization, with results composited using standard alpha blending. The four experimental conditions are illustrated in Figure 1.

Hypotheses. Our method is designed to increase both class cohesion and color discriminability. Hence, we expect our optimization to improve user performance in the two tasks:

H1 – In the *Distribution Estimation* task, we anticipate our method to improve the perception of distribution features over the benchmarks. Therefore, we expect participants to perform better shape estimation with *CNA* than with *Standard*, *Local*, and *Hue-preserving*.

H2 – In the *Class Discrimination* task, we similarly predict our method will improve participants' performance in class

discrimination, outperforming a fixed color assignment strategy (*Standard*, *Local*, and *Hue-preserving*).

Furthermore, we anticipate the optimized colorization from our approach to result in an aesthetically pleasing color scheme:

H3 – Participants will show comparable preference for *CNA* and the three benchmarks (*Standard*, *Local*, *Hue-preserving*).

Experiment Design. The task was varied between subjects, with different participants recruited for each of the three tasks. The experimental condition was a within-subject variable, with each participant experiencing all four colorization methods. To minimize potential learning effects, we displayed the stimuli in a random order. We recruited participants from Prolific, which is shown to generate higher quality responses as compared with other crowdsourcing platforms [11]. We limited recruitment to residents of the UK and US, while balancing the proportion of male and female through Prolific recruitment options.

To ensure the accuracy of the result, we asked participants to complete a color deficiency test at the beginning of the experiment and randomly inserted multiple engagement checks within the stimulus sequence. These checks featured two completely separated Gaussian distributions in easily distinguishable colors, making them straightforward to judge. Participants who did not correctly resolve these checks were excluded from the analysis.

After the participants passed the color deficiency test, they were presented with task instructions and completed three training trials. Following the training phase, they proceeded to complete the analyzed trials, which consisted of either 72 stimuli (18 histograms × 4 methods) for the distribution estimation task and class discrimination task, or 54 stimuli (18 histograms × 3 pairs) for the user preference task. Each stimulus had a response time limit of 30 seconds, after which participants would be directed to the next trial.

1) Results for Distribution Estimation: A total of 40 participants (18 males, 21 females, and 1 unspecified) were

included in the analysis for this task. We initially excluded five participants who failed the attention checks, and replaced them with new recruits to reach our intended sample size of 40 responses. On average, participants completed the experiment in 7.20 minutes ($\sigma = 3.37$). We set the compensation at $1.5 to exceed the US minimum hourly wage. Our analysis includes 95% confidence intervals, effect sizes (Cohen's d) for both error rate and response time, and p-values derived from Mann-Whitney tests (based on a positive Shapiro-Wilk test). Given the non-normality of the error data, we employed the GFD R-package [14] to compute ANOVA-type interaction statistics.

Figure 7 illustrates the results for this task. We find that our proposed method (CNA) outperforms the benchmarks, which feature fixed color and opacity assignment (*Standard*, *Local*, *Hue-preserving*), both in terms of error rate and response time. Notably, our method demonstrates a significantly lower error rate than all three benchmark conditions. We found an interaction between the colorization methods and the class number ($p = 0.041$) for the error rate. However, there was no significant interaction between the experimental conditions and smoothness ($p = 0.244$). Hence, while the number of classes seems to affect the tested methods somewhat differently, the smoothness of the distribution appears to have a consistent effect, with unsmooth distributions uniformly more challenging across all tested techniques, including ours (see Figure 10).

We analyzed response times by first removing extreme outliers to reveal more systematic effects for the visualization technique. This was done by pruning participant trials that are ±2 standard deviations away from their mean response time. There was no significant difference in response time between our approach and the three benchmarks, suggesting that our optimization does not significantly change response time. These results are consistent with **H1**.
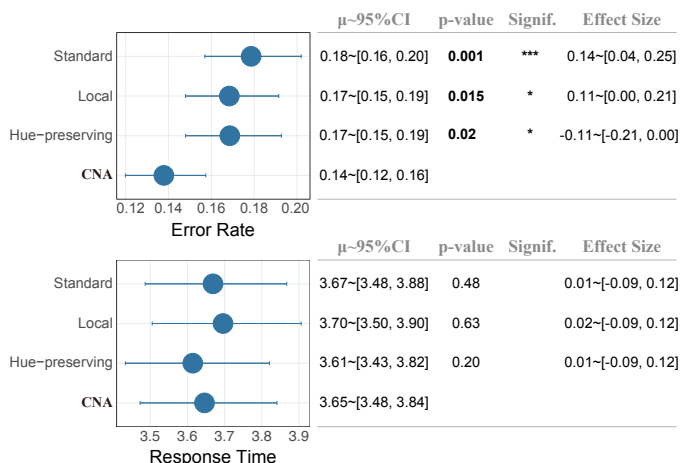

	μ~95%CI	p-value	Signif.	Effect Size
Standard	0.18~[0.16, 0.20]	**0.001**	***	0.14~[0.04, 0.25]
Local	0.17~[0.15, 0.19]	**0.015**	*	0.11~[0.00, 0.21]
Hue-preserving	0.17~[0.15, 0.19]	**0.02**	*	-0.11~[-0.21, 0.00]
CNA	0.14~[0.12, 0.16]			

	μ~95%CI	p-value	Signif.	Effect Size
Standard	3.67~[3.48, 3.88]	0.48		0.01~[-0.09, 0.12]
Local	3.70~[3.50, 3.90]	0.63		0.02~[-0.09, 0.12]
Hue-preserving	3.61~[3.43, 3.82]	0.20		0.01~[-0.09, 0.12]
CNA	3.65~[3.48, 3.84]			

Fig. 8.   Results for the *Class Discrimination Task* from Study I.

2) Results for Class Discrimination: We similarly recruited 40 participants (20 males, 20 females) for this task. Three participants were initially excluded for failing attention checks, with additional participants recruited in replacement. Each participant completed a total of 72 stimuli, with an average completion time of 5.26 minutes ($\sigma = 3.62$) for the entire ex-

periment. We compensated participants with a $1.2 payment to exceed the US hourly minimum wage. The results are depicted in Figure 8. The error rate with our optimization method was consistently lower than that of the fixed assignment strategy. Specifically, our optimization achieved a significantly reduced error rate compared to the benchmarks (*Standard*, *Local*, *Hue-preserving*). These results are consistent with **H2**. We found no significant interaction effect between the experimental condition and the number of histogram classes ($p = 0.267$), or with the distribution smoothness ($p = 0.575$). This suggests that all tested techniques are similarly affected by the complexity of the visualization. In terms of response time, we found no significant difference between the four methods, even CNA led to a faster response, on average.

3) Results for User Preference: We enrolled 40 participants (20 males, and 20 females, with no participants excluded). Participants received a compensation of $0.75. For each comparison, we assigned a score of 1 if the participant preferred our approach and -1 for preferring one of the other benchmarks. Zero was assigned for a neutral choice when the participant indicated no clear preference. The results are depicted in Figure 9, with a positive score indicating a preference for our technique. Our method (*CNA*) appears preferable to *Hue-preserving* ($p = 0.06$). Other comparisons did not reach statistical significance. We estimated the difference in preference between our technique relative to Standard (0.02, 95% CI: $[-0.05, 0.08]$) and Local blending (-0.01, 95% CI: $[-0.09, 0.05]$). In both cases, confidence intervals are centered approximately around zero, suggesting that *CNA* is comparable to the benchmarks, if not more preferable. These results are consistent with **H3**.

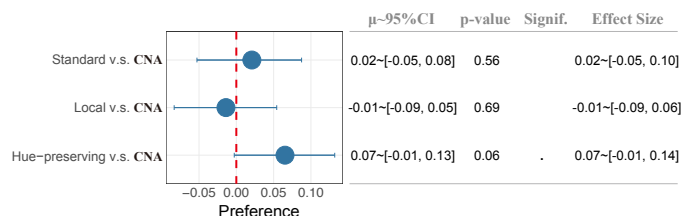|	μ~95%CI	p-value	Signif.	Effect Size
Standard v.s. **CNA**	0.02~[-0.05, 0.08]	0.56		0.02~[-0.05, 0.10]
Local v.s. **CNA**	-0.01~[-0.09, 0.05]	0.69		-0.01~[-0.09, 0.06]
Hue-preserving v.s. **CNA**	0.07~[-0.01, 0.13]	0.06	.	0.07~[-0.01, 0.14]

Fig. 9.   Results for the *User Preference Task* from Study I. A preference score greater than 0 indicates that participants prefer our method over the respective benchmark.

4) Relationship between Smoothness and Error: To further understand the role of distribution smoothness, we analyzed the relationship between the smoothness metric defined in §IV and the observed empirical error for the distribution estimation task (results for the class discrimination are similar). As expected, we found a significant positive correlation, with unsmooth distribution being significantly more difficult to interpret by participants ($F(1, 70) = 49.34; p < 0.001$), likely due to increased visual complexity. This correlation with error is slightly decreased with our optimization as compared to local and standard blending (see Figure 10), although the interaction is not significant.

B. Study II: Histograms + Kernel Density Curves

Results from Study I suggest that our optimization can enhance the perception of semi-transparent histograms, ef-
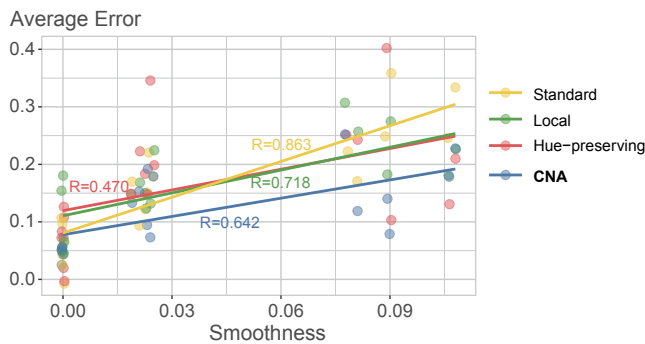
Fig. 10. Effect of smoothness on error in the distribution estimation task. Higher x-values indicate less smooth distributions. The R-value represents Pearson's correlation coefficient between the smoothness value and the average error rate. Each point represents one of the 18 histograms with certain condition (i.e., 18*4=72 points in total).
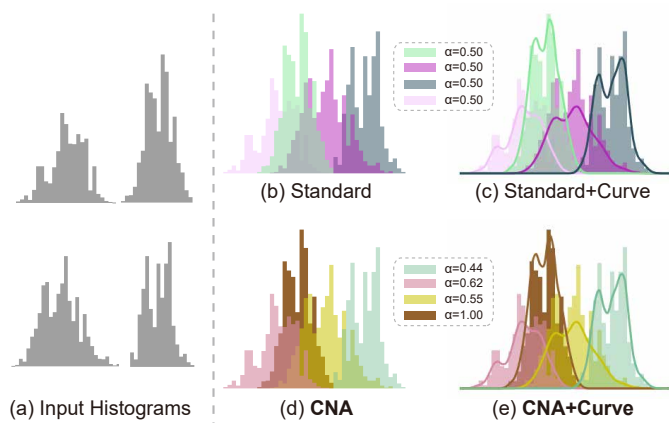

Fig. 11. Four example stimuli from Study II showing histograms with and without a kernel density estimate curve. (a) The input distributions with four classes. The top row (b, c) represents standard alpha blending with a palette generated via Colorgorical. The bottom row (d, e) shows the results of our optimization.

fectively compensating for the complex interaction between translucent colors. However, in practice, many visualization designers opt to augment these histograms with different visual representations, such as a curve representing kernel density estimate [66], or a hatching pattern [22]. This additional visual cue can mitigate the challenges associated with translucent marks. We thus conducted a second study about the density curve, a most commonly used technique, to assess whether our optimization can still provide an advantage in this scenario.

Conditions, Stimuli, and Experiment Design. Our primary focus in this study was the *distribution estimation task*, which serves as a proxy for distribution analysis. Building on the findings from Study I, we tested standard alpha blending against our method, adding two new additional variations: standard alpha blending with density curve (abbreviated as *Standard with Curve*) and our method plus density curve (*CNA with Curve*). These four conditions are illustrated in Figure 11. The experiment followed a within-subjects design, with all participants experiencing the four conditions above. Each participant completed a total of 72 trials, representing a factorization of 4 conditions × 18 histograms. The order of presentation for the conditions was randomized.

Hypotheses. We expected the addition of a kernel density curve to improve participant performance in all four conditions. However, we anticipate the benefit to be maximum when combined with our optimization. Thus, we pose the following hypotheses:

H4 – We anticipate that the addition of a density curve will significantly improve user performance, no matter the condition. As a result, we expect *Standard with Curve* to exhibit better performance than *CNA* (i.e., a plain condition with no density cues).

H5 – Our optimization with combined density curve will outperform all other conditions.

Participants & Procedure. For this study, we recruited 40 participants (20 males, and 20 females) from the Prolific. To exceed the US minimum wage, we compensated each participant with $1.5. All participants passed the color vision deficiency test. The experimental procedure closely mirrored that of the previous study.

Results. The average completion time for participants was 6.94 minutes ($\sigma = 2.56$). The experiment results are shown in Figure 12. Our optimization, both with and without the inclusion of density curves (*CNA* and *CNA with Curve*), afforded higher average accuracy compared to using a fixed color assignment strategy (*Standard* and *Standard with Curve*). This result suggests that even with the inclusion of enhancing visual cues, a conventional plot featuring overlapping histograms will not surpass the effectiveness of our technique. These results, however, are inconsistent with **H4**. That said, combining our method with a density curve (*CNA with Curve*) appears to further reduce the error rate relative to the baseline technique (*CNA*), yielding significantly lower error rates than both *Standard* and *Standard with Curve* ($p < 0.01$). These results support **H5**. This suggests that our method offers an additional advantage beyond what can be attained with histogram-enhancing kernel density estimates. There was no significant interaction between the representation method and the number of classes ($p = 0.251$), or with distribution smoothness ($p = 0.593$), suggesting consistent effects for the latter two factors.

As for response time, we found that our optimization without a density curve (*CNA*) led to a significantly faster response than *Standard with Curve* ($p < 0.001$) and *CNA with Curve* ($p < 0.001$). This suggests that adding a kernel density estimate increases the complexity of the visualization, potentially requiring more time for the viewer to parse it. We further looked for evidence of a correlation between error rates and response time to understand tradeoffs in speed vs. accuracy. we found that there is generally no significant correlation between the two (see Figure 13). While *Standard with Curve* exhibits some negative correlation between error and response time as predicted ($R = -0.259$), this correlation is not statistically significant. A potential explanation for this result is that participants were instructed to respond as accurately as possible, leading them to allocate more time to each stimulus, thereby reducing variability in response times.

	μ~95%CI	p-value	Effect Size	p-value	Effect Size
Standard	0.17~[0.15, 0.20]	0.31	0.04~[-0.06, 0.15]	**0.01**	0.11~[0.01, 0.22]
Standard with Curve	0.17~[0.15, 0.20]	0.27	0.05~[-0.05, 0.15]	**0.01**	0.12~[0.01, 0.22]
CNA	0.16~[0.13, 0.18]	CNA as baseline		0.13	0.07~[-0.04, 0.17]
CNA with Curve	0.13~[0.11, 0.16]	0.13	0.07~[-0.04, 0.17]	CNA with Curve as baseline	

Error Rate

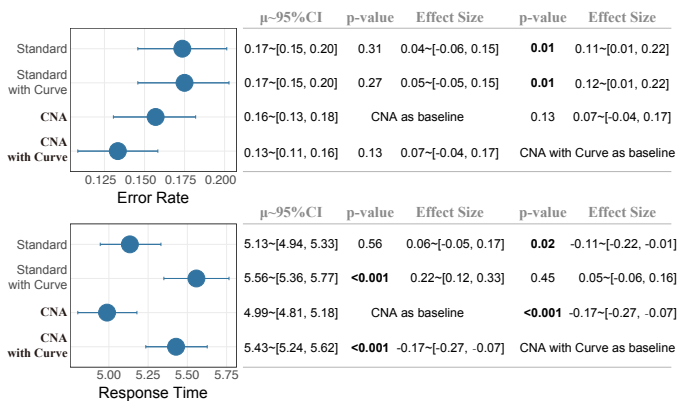|	μ~95%CI	p-value	Effect Size	p-value	Effect Size
Standard	5.13~[4.94, 5.33]	0.56	0.06~[-0.05, 0.17]	**0.02**	-0.11~[-0.22, -0.01]
Standard with Curve	5.56~[5.36, 5.77]	**<0.001**	0.22~[0.12, 0.33]	0.45	0.05~[-0.06, 0.16]
CNA	4.99~[4.81, 5.18]	CNA as baseline		**<0.001**	-0.17~[-0.27, -0.07]
CNA with Curve	5.43~[5.24, 5.62]	**<0.001**	-0.17~[-0.27, -0.07]	CNA with Curve as baseline	

Response Time

Fig. 12. Results for the *Distribution Estimation Task* in Study II. The top shows accuracy results whereas the bottom chart depicts response time. P-values and effects reflect comparisons with *CNA* and *CNA with Curve*, respectively.
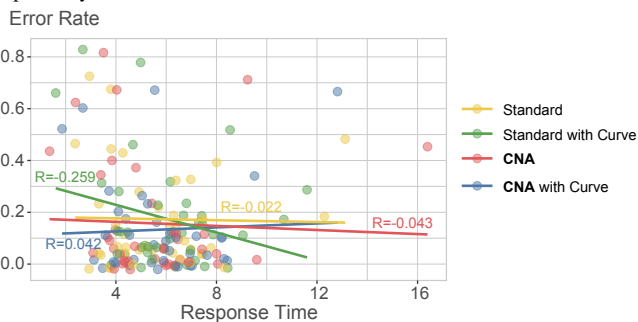

Fig. 13. Effect of response time on error in Study II. The R-value represents Pearson's correlation coefficient between the response time and the error rate for the *Distribution Estimation Task*. Points represent mean participant performance in one of the four conditions.

C. Discussion

Our findings suggest that the proposed method offers significant benefits for interpreting complex, translucent histogram plots, even when compared to blending methods that are designed for transparency conditions. In the first study, we specifically compared our method to three other benchmarks representing a standard alpha blending approach and two more specialized methods for preserving hues and local color properties. Findings from the two analytical tasks show that our optimization led participants to outperform the alternatives, providing support for H1 and H2. Notably, the hue-preserving color blending model (*Hue-preserving*) exhibited the poorest performance in both tasks. This subpar performance is likely attributable to the generation of neutral gray colors in the overlapping areas, which could potentially lead to a breakdown in color nameability. By contrast, our optimization reinforces color-name associations between the overlapping segments and their original class color.

Our approach similarly outperformed a local blending model, intended to enhance depth perception by reducing the saturation of 'deeper' objects. However, this approach may impede the perception of wholes from parts in overlapping distributions. In contrast, enhancing color name similarity across all parts of a histogram appears to be a more appropriate strategy. Our technique leverages color nameability while also guaranteeing perceptual discriminability against both the background and other overlapping distributional features within the visualizations. Such optimization does not seem to diminish the aesthetic appeal of the visualization (consistent with H3). Indeed, participants expressed a comparable preference for our technique compared to the benchmark.

In the second study, we evaluated the perception of histograms augmented with curves representing kernel density estimates. Here, our optimization outperformed a straightforward approach of using fixed color and opacity assignments, even when the effects were not always significant. This finding extends the results of the first study, indicating that our technique can effectively substitute kernel density curves to enhance perception. Thus, our approach could be preferable when designers want to maintain the simplicity of the visualization without introducing additional cues (H4 not confirmed). Moreover, for an even greater performance improvement, combining a density estimate with our optimization results in the highest accuracy (H5 confirmed). To illustrate this effect, consider the pink distribution in Figure 11-a. This visualization remains challenging to interpret even with the addition of a density curve (b). However, when colors and transparency are optimized using our method, the same histogram becomes easier to discern. Adding a density outline (d) yields to the best empirically observed performance.

D. Study Limitations

There are some limitations to our experiments that should be contextualized. First, the diversity of our histogram datasets may be limited. Although we attempted to vary stimuli characteristics, such as the number of classes, the degree of overlap, and the smoothness of the distributions, other factors could also influence perception. These include the distribution family (e.g., Gaussian vs. exponential). Future work could study the impact of these factors. Moreover, there are additional techniques for improving the readability of the translucent histograms, including stepped histograms [9] and hatching pattern [22]. While our study employs commonly used techniques, future work could compare or combine our optimization with other representations. Lastly, our evaluation was limited to a shape estimation task for a single distribution. Future work should investigate the potential of our method in more complex, comparative tasks, such as comparing shapes or features across multiple distributions [7].

V. Extensions

Since our optimization is computed in the screen space, it can be readily applied to other chart types, such as illustrative parallel coordinates and Venn diagrams. This is done by first rendering each layer in a separate canvas to obtain the corresponding pixel layout. We then optimize color assignment and opacity with our algorithm, considering the overlap between the layers, their neighborhood graph, and the size of each color segment. To demonstrate the effectiveness of our technique in these scenarios, we present case studies with real-world datasets.
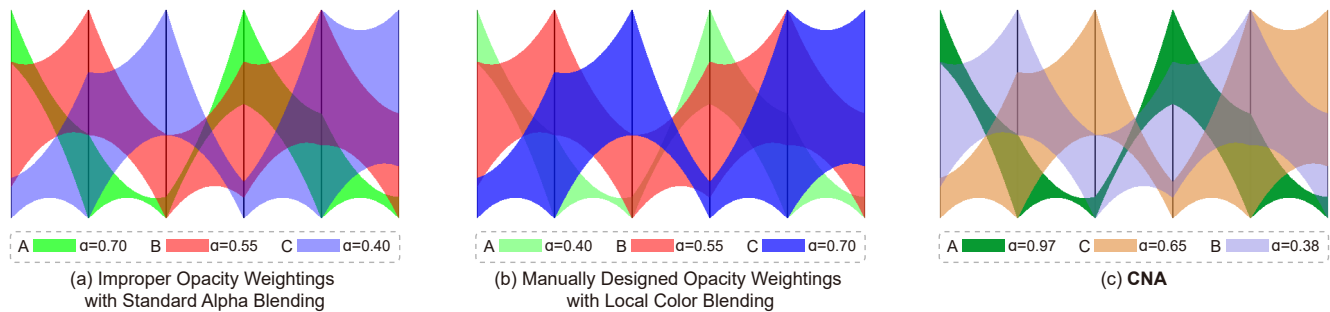
Fig. 14. Visualizing the Scan Bio dataset with illustrative parallel coordinates. results generated by using (a) default opacity weightings and standard alpha blending, (b) local color blending, and (c) our optimization.

A. Illustrative Parallel Coordinates

We visualize six dimensions of the ScanBio dataset[2] with illustrative parallel coordinates [41]. There are three layers to this chart colorized as green, red, and blue ribbons (see Figure 14). We show this visualization with default colors and transparency as appearing in the original work [41]. We compare with local color blending and our optimization. Viewed with default opacity settings using standard alpha blending the result introduces new false colors, including what might be interpreted as a purple class. Using the suggested opacity weightings derived with local color blending (b), the 'depth' of the chart is perceived more clearly, but the overlapping segments seem difficult to discriminate individually. Our auto-generated color and transparency levels (c) yield a qualitatively improved visualization, enhancing the ribbon's visual continuity while still allowing the identification of overlapping segments.
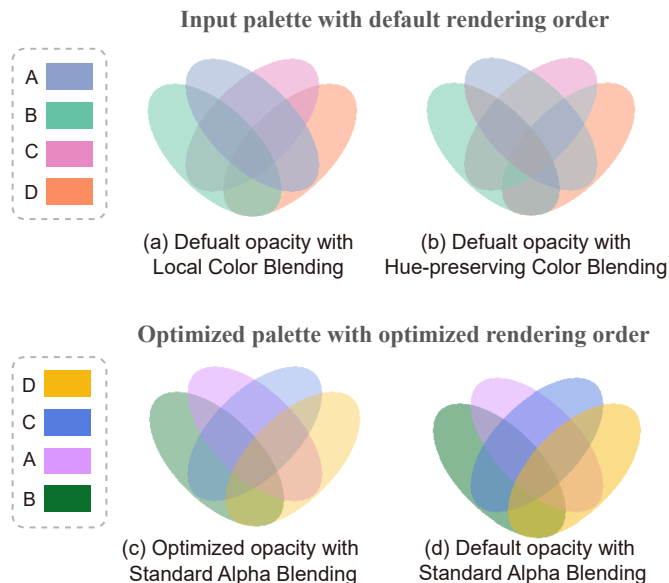

Fig. 15. An elliptical Venn diagram shown with (a) ColorBrewer palette and uniform opacity (0.5) using local blending, and (b) hue-preserving blending. Both visualizations lead to suboptimal compositing. By contrast, our optimization (c) produces a more perceptible set visualization which guarantees minimum color separability for all parts. (d) Shows the same colors as (c) but with uniform, non-optimized opacities.

[2]Obtained from http://davis.wpi.edu/~xmdv.

B. Venn Diagrams

Venn diagrams are extensively utilized for visualizing sets [26], providing an intuitive representation to identify set intersections, unions, and differences. Incorporating color enhances the perception of set boundaries, but color interactions could also compromise the diagram's effectiveness [39]. To illustrate the applicability of our method in this domain, we generated a Venn diagram with four intersecting elliptical sets using Venny [45]. We colorized the ellipses using a 4-class palette generated from ColorBrewer (uniform opacity of $\alpha = 0.5$), and then applied local blending model and hue-preserving model. Results are shown in Figure 15. Observe how the intersection between the blue and red sets is challenging to associate with the red component. Similarly, employing a hue-preserving color blending model generates an abundance of grey tones. With our approach (c), the resulting color and transparency appear to qualitatively improve the visualization. For instance, the overlapping parts of the green set are more readily identifiable as green. This visualization can overall be read with minimal ambiguity.

VI. CONCLUSION AND FUTURE WORK

We proposed a color-name aware optimization approach that automatically generates color assignment and determines proper transparency settings for overlapping histograms. Compared to existing blending models, our approach allows for a more accurate estimation of distributions and color classes. We achieved these results by optimizing for both within-class name association and between-class disassociation, and by also considering region-based color contrast. These factors are scored using an objective function and optimized using a custom simulated annealing algorithm. Our method allows for rapidly generating new color palettes and transparency assignments to reduce misinterpretation errors. We demonstrated the effectiveness of our method in two experiments. We also show extensions to other multi-class visualizations with overlaps, such as illustrative parallel coordinates and Venn diagrams. To help disseminate this method, we implemented our technique in a web-based tool that integrates our optimization.

Although effective, the quality of our optimization may be influenced by the color naming model used. In this study, we utilized the Heer and Stone name model [25], which at times showed inconsistent accuracy. For example, the name distance between bright hues (e.g. light blue and light green)

displayed higher variation than expected. Future research could explore alternative color naming models. Another limitation lies in the variability of the algorithm's results across different runs of simulated annealing, which stems from its inherent stochastic nature. To address this issue in future work, alternative optimization algorithms, such as genetic algorithms or particle swarm optimization, could be explored to yield deterministic solutions. Another limiting factor is the need to account for the effect of the algorithm on emotional valence. Studies suggest that individual colors are often associated with specific emotional responses [4], [5], whereas blended colors can alter, amplify, or even give rise to entirely different emotional associations. To address this challenge, future work should incorporate the evaluation of emotional responses to color blending, ensuring that visualization designs are both semantically resonant and emotionally appropriate. Additionally, we aim to expand our optimization framework to generate color palettes friendly to color vision deficiency (CVD). Our method can be further extended to 3D scenarios, such as applications involving Gaussian Mixture Models [31], by incorporating relevant visual cues and spatial properties. Finally, we would like to more thoroughly investigate how data characteristics (e.g., distribution family and shape) might impact task accuracy.

ACKNOWLEDGMENTS

REFERENCES

[1] E. Aarts. A stochastic approach to combinatorial optimization and neural computing. *Simulated Annealing and Boltzmann Machines*, 1989.

[2] E. H. Adelson and P. Anandan. *Ordinal characteristics of transparency*. Citeseer, 1990.

[3] D. Albers, M. Correll, and M. Gleicher. Task-driven evaluation of aggregation in time series visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 551–560, 2014. doi: 10.1145/2556288.2557200

[4] C. L. Anderson and A. C. Robinson. Affective congruence in visualization design: Influences on reading categorical maps. *IEEE Transactions on Visualization and Computer Graphics*, 28(8):2867–2878, 2021. doi: 10.1109/TVCG.2021.3050118

[5] L. Bartram, A. Patra, and M. Stone. Affective color in visualization. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 1364–1374, 2017. doi: 10.1145/3025453.302604

[6] J. Beck, K. Prazdny, and R. Ivry. The perception of transparency with achromatic colors. *Perception & psychophysics*, 35(5):407–422, 1984.

[7] M. Blumenschein, L. J. Debbeler, N. C. Lages, B. Renner, D. A. Keim, and M. El-Assady. v-plots: Designing hybrid charts for the comparative analysis of data distributions. In *Computer Graphics Forum*, vol. 39, pp. 565–577. Wiley Online Library, 2020. doi: 10.1111/cgf.14002

[8] J. Chuang, D. Weiskopf, and T. Moller. Hue-preserving color blending. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1275–1282, 2009. doi: 10.1109/TVCG.2009.150

[9] S. Y. Claire, K. Kremer, S. Chatterjee, C. L. Rodriguez, and F. A. Rasio. Millisecond pulsars and black holes in globular clusters. *The Astrophysical Journal*, 877(2):122, 2019. doi: 10.3847/1538-4357/ab1b21

[10] M. Correll, M. Li, G. Kindlmann, and C. Scheidegger. Looks good to me: Visualizations as sanity checks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):830–839, 2018. doi: 10.1109/TVCG.2018.2864907

[11] B. D. Douglas, P. J. Ewell, and M. Brauer. Data quality in online human-subjects research: Comparisons between mturk, prolific, cloudresearch, qualtrics, and sona. *Plos one*, 18(3):e0279720, 2023. doi: 10.1371/journal.pone.0279720

[12] M. El-Assady, R. Kehlbeck, Y. Metz, U. Schlegel, R. Sevastjanova, F. Sperrle, and T. Spinner. Semantic color mapping: A pipeline for assigning meaningful colors to text. In *2022 IEEE 4th Workshop on Visualization Guidelines in Research, Design, and Education (VisGuides)*, pp. 16–22. IEEE, 2022.

[13] J. D. Foley. *Computer graphics: principles and practice*, vol. 12110. Addison-Wesley Professional, 1996.

[14] S. Friedrich, F. Konietschke, and M. Pauly. GFD: An R package for the analysis of general factorial designs. *Journal of Statistical Software, Code Snippets*, 79(1):1–18, 2017. doi: 10.18637/jss.v079.c01

[15] S. Gama and D. Gonçalves. Guidelines for using color blending in data visualization. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 363–364, 2014. doi: 10.1145/2598153.2600039

[16] S. Gama and D. Goncalves. Studying color blending perception for data visualization. 2014. doi: 10.2312/eurovisshort.20141168

[17] D. Gigilashvili, J.-B. Thomas, J. Y. Hardeberg, and M. Pedersen. Translucency perception: A review. *Journal of Vision*, 21(8):4–4, 2021.

[18] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 21 pages, 2011. doi: 10.1177/1473871611416549

[19] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss. Colorgorical: creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):521–530, 2017. doi: 10.1109/TVCG.2016.2598918

[20] T. Günther, C. Rössl, and H. Theisel. Opacity optimization for 3d line fields. *ACM Transactions on Graphics (TOG)*, 32(4):1–8, 2013. doi: 10.1145/2461912.2461930

[21] T. Günther, M. Schulze, J. M. Esturo, C. Rössl, and H. Theisel. Opacity optimization for surfaces. In *Computer Graphics Forum*, vol. 33, pp. 11–20. Wiley Online Library, 2014. doi: 10.1111/cgf.12357

[22] B. Gutmann and T. Stelzer. Values affirmation replication at the university of illinois. *Physical Review Physics Education Research*, 17(2):020121, 2021. doi: 10.1103/PhysRevPhysEducRes.17.020121

[23] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1270–1277, 2007. doi: 10.1109/TVCG.2007.70623

[24] M. Harrower and C. A. Brewer. ColorBrewer.org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003. doi: 10.1179/000870403235002042

[25] J. Heer and M. Stone. Color naming models for color selection, image editing and palette design. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1007–1016, 2012. doi: 10.1145/2207676.2208547

[26] A. Jia, L. Xu, and Y. Wang. Venn diagrams in bioinformatics. *Briefings in bioinformatics*, 22(5):bbab108, 2021. doi: 10.1093/bib/bbab108

[27] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 125–132. IEEE, 2005. doi: 10.1109/INFVIS.2005.1532138

[28] J. J. Kaplan, J. G. Gabrosek, P. Curtiss, and C. Malone. Investigating student understanding of histograms. *Journal of Statistics Education*, 22(2), 2014. doi: 10.1080/10691898.2014.11889701

[29] L. Kühne, J. Giesen, Z. Zhang, S. Ha, and K. Mueller. A data-driven approach to hue-preserving color-blending. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2122–2129, 2012. doi: 10.1109/TVCG.2012.186

[30] E. Languenou. Contrast driven color-group assignment in categorical data visualization. 2023. doi: 10.5220/0011615000003417

[31] K. Lawonn, M. Meuschke, P. Eulzer, M. Mitterreiter, J. Giesen, and T. Günther. Gray: Ray casting for visualization and interactive data exploration of gaussian mixture models. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):526–536, 2023. doi: 10.1109/TVCG.2022.3209374

[32] S. Lem, P. Onghena, L. Verschaffel, and W. Van Dooren. On the misinterpretation of histograms and box plots. *Educational Psychology*, 33(2):155–174, 2013. doi: 10.1080/01443410.2012.674006

[33] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. *Computer Graphics Forum*, 32(3):401–410, 2013. doi: 10.1111/cgf.12127

[34] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang. Palettailor: discriminable colorization for categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):475–484, 2021. doi: 10.1109/TVCG.2020.3030406

[35] K. Lu, K. Reda, O. Deussen, and Y. Wang. Interactive context-preserving color highlighting for multiclass scatterplots. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2023. doi: 10.1145/3544548.3580734

[36] Y. Lu, L. Cheng, T. Isenberg, C.-W. Fu, G. Chen, H. Liu, O. Deussen, and Y. Wang. Curve complexity heuristic kd-trees for neighborhood-based exploration of 3d curves. In *Computer Graphics Forum*, vol. 40, pp. 461–474. Wiley Online Library, 2021. doi: 10.1111/cgf.142647

[37] M. Luboschik, A. Radloff, and H. Schumann. A new weaving technique for handling overlapping regions. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pp. 25–32, 2010. doi: 10.1145/1842993.1842999

[38] M. Mahy, L. Van Eycken, and A. Oosterlinck. Evaluation of uniform color spaces developed after the adoption of cielab and cieluv. *Color Research & Application*, 19(2):105–121, 1994. doi: 10.1111/j.1520-6378.1994.tb00070.x

[39] K. Mann. The calcified eggshell matrix proteome of a songbird, the zebra finch (taeniopygia guttata). *Proteome science*, 13:1–20, 2015. doi: 10.1186/s12953-015-0086-1

[40] J. Matejka, F. Anderson, and G. Fitzmaurice. Dynamic opacity optimization for scatter plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2707–2710, 2015. doi: 10.1145/2702123.2702585

[41] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. In *Computer Graphics Forum*, vol. 27, pp. 1031–1038. Wiley Online Library, 2008. doi: 10.1111/j.1467-8659.2008.01239.x

[42] F. Metelli. The perception of transparency. *Scientific American*, 230(4):90–99, 1974. doi: 10.1038/scientificamerican0474-90

[43] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauf. Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1588–1599, 2017. doi: 10.1109/TVCG.2017.2674978

[44] K. Mukherjee, B. Yin, B. E. Sherman, L. Lessard, and K. B. Schloss. Context matters: A theory of semantic discriminability for perceptual encoding systems. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):697–706, 2021.

[45] J. C. Oliveros. Venny. an interactive tool for comparing lists with venn diagrams. *http://bioinfogp. cnb. csic. es/tools/venny/index. html*, 2007.

[46] T. Porter and T. Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 253–259, 1984. doi: 10.1145/800031.808606

[47] G. J. Quadri, J. A. Nieves, B. M. Wiernik, and P. Rosen. Automatic scatterplot design optimization for clustering identification. *IEEE Transactions on Visualization and Computer Graphics*, 2022. doi: 10.1109/TVCG.2022.3189883

[48] K. Reda. Rainbow colormaps: What are they good and bad for? *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5496–5510, 2022.

[49] K. Reda, A. A. Salvi, J. Gray, and M. E. Papka. Color nameability predicts inference accuracy in spatial visualizations. In *Computer Graphics Forum*, vol. 40, pp. 49–60. Wiley Online Library, 2021. doi: 10.1111/cgf.14288

[50] K. Reda and D. A. Szafir. Rainbows revisited: Modeling effective colormap design for graphical inference. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1032–1042, 2020.

[51] riyaaggarwal. Overlapping histograms with matplotlib in python. https://www.geeksforgeeks.org/overlapping-histograms-with-matplotlib-in-python/, 2020. Accessed: 2020-11-26.

[52] R. Sahann, T. Müller, and J. Schmidt. Histogram binning revisited with a focus on human perception. In *2021 IEEE Visualization Conference (VIS)*, pp. 66–70. IEEE, 2021. doi: 10.1109/VIS49827.2021.9623301

[53] K. B. Schloss, Z. Leggon, and L. Lessard. Semantic discriminability for visual communication. *Ieee transactions on visualization and computer graphics*, 27(2):1022–1031, 2020.

[54] K. B. Schloss, L. Lessard, C. S. Walmsley, and K. Foley. Color inference in visual communication: the meaning of colors in recycling. *Cognitive research: principles and implications*, 3:1–17, 2018.

[55] V. Setlur and M. C. Stone. A linguistic approach to categorical color assignment for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):698–707, 2016. doi: 10.1109/TVCG.2015.2467471

[56] G. Sharma, W. Wu, and E. N. Dalal. The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005. doi: 10.1002/col.20070

[57] M. Singh and B. L. Anderson. Toward a perceptual theory of transparency. *Psychological review*, 109(3):492, 2002.

[58] T. Software. The tableau visualization system. http://www.tableausoftware.com/.

[59] D. A. Szafir. Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):392–401, 2018. doi: 10.1109/TVCG.2017.2744359

[60] T. Urness, V. Interrante, I. Marusic, E. Longmire, and B. Ganapathisubramani. Effectively visualizing multi-valued flow data using color and texture. In *IEEE Visualization, 2003. VIS 2003.*, pp. 115–121. IEEE, 2003. doi: 10.1109/VISUAL.2003.1250362

[61] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1739–1754, 2008. doi: 10.1109/TVCG.2008.118

[62] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C.-W. Fu, O. Deussen, and B. Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):820–829, 2019. doi: 10.1109/TVCG.2018.2864912

[63] Y. Wang, J. Zhang, W. Chen, H. Zhang, and X. Chi. Efficient opacity specification based on feature visibilities in direct volume rendering. In *Computer Graphics Forum*, vol. 30, pp. 2117–2126. Wiley Online Library, 2011. doi: 10.1111/j.1467-8659.2011.02045.x

[64] D. Weiskopf et al. *GPU-based interactive visualization techniques*. Springer, 2007.

[65] M. Wertheimer. Gestalt theory. 1938.

[66] C. O. Wilke. *Fundamentals of data visualization: a primer on making informative and compelling figures*. O'Reilly Media, 2019.

[67] Y. Yang, J. Ming, and N. Yu. Color image quality assessment based on ciede2000. *Advances in Multimedia*, 2012:11–11, 2012. doi: 10.1155/2012/273723

[68] L. Zheng, Y. Wu, and K.-L. Ma. Perceptually-based depth-ordering enhancement for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):446–459, 2012. doi: 10.1109/TVCG.2012.144

[69] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. In *Computer Graphics Forum*, vol. 27, pp. 1047–1054. Wiley Online Library, 2008. doi: 10.1111/j.1467-8659.2008.01241.x

Kecheng Lu is a postdoctoral researcher at the School of Information, Renmin University of China. He completed his Ph.D. in Computer Science at Shandong University in 2023. His research focuses on information visualization and human-computer interaction.

Lihang Zhu is currently a second-year Master student in School Of Computer Science and Technology, Shandong University. His research interests include information visualization and human computer interaction.
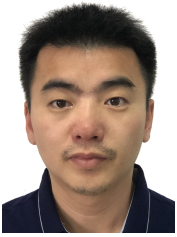
Yunhai Wang is a professor in School of Information, Renmin University of China. He serves as the associate editor of IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Graphics and Applications, and Computer Graphics Forum. His interests include scientific visualization, information visualization, and computer graphics.

Qiong Zeng is an associate professor in the School of Computer Science and Technology at Shandong University, China. Her research interests include scientific visualization and computer graphics. Qiong obtained her Ph.D. in Software Engineering from Shandong University, China, in 2015. She is a member of the IEEE Computer Society. Contact her at qiong.zn@sdu.edu.cn.

Weitao Song received the BSc and PhD degrees in optics from the Beijing Institute of Technology, China, in 2010 and 2016, respectively. He is currently a professor with the School of Optics and Photonics, Beijing Institute of Technology, China. His research interests include color science, optical instrumentation, virtual reality, and augmented reality.

Khairi Reda is an Associate Professor in the School of Informatics and Computing at Indiana University Indianapolis. His research lies at the intersection of Human-Computer Interaction and Data Science, with interests in visual analytics and data cognition.