Cieran: Designing Sequential Colormaps via In-Situ Active Preference Learning

Matt-Heun Hong
Department of Computer Science
University of North Carolina
at Chapel Hill
Chapel Hill, NC, USA
mhh@cs.unc.edu

Zachary Nolan Sunberg
Smead Aerospace Engineering
Sciences Department
University of Colorado Boulder
Boulder, CO, USA
zachary.sunberg@colorado.edu

Danielle Albers Szafir
Department of Computer Science
University of North Carolina
at Chapel Hill
Chapel Hill, NC, USA
danielle.szafir@cs.unc.edu

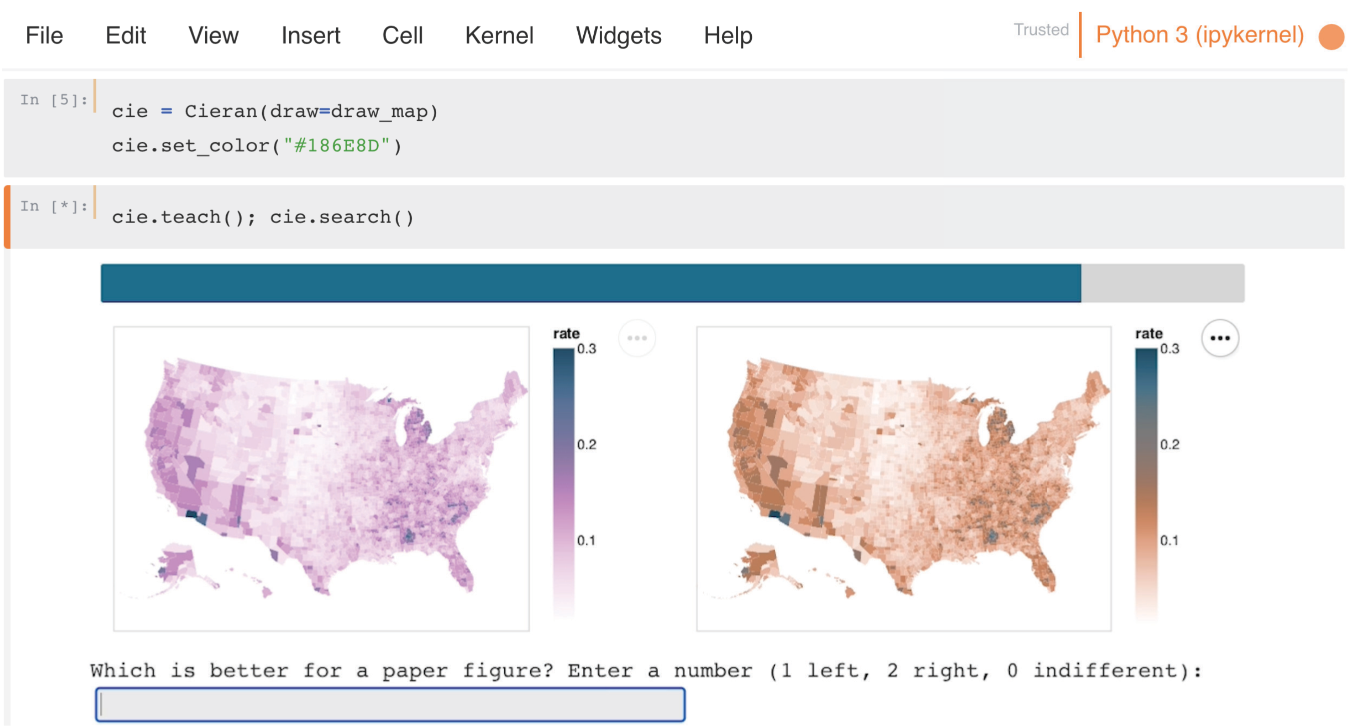

Figure 1: Cieran's preference learning interface used with Altair [71]. Cieran supports efficient colormap selection within an analyst's workflow through integration with Jupyter Notebooks. Cieran first interpolates example colormaps through a chosen color (e.g., #186E8D, a teal blue shown in the progress bar). People iteratively input preferences to Cieran by making value judgements across pairs of examples, and Cieran uses the preference data to induce a context-specific model of aesthetic utility. This model used to rank and create colormaps.

ABSTRACT

Quality colormaps can help communicate important data patterns. However, finding an aesthetically pleasing colormap that looks "just right" for a given scenario requires significant design and technical expertise. We introduce Cieran, a tool that allows any data analyst to rapidly find quality colormaps while designing charts within Jupyter Notebooks. Our system employs an active preference learning paradigm to rank expert-designed colormaps and create new ones from pairwise comparisons, allowing analysts who are novices in color design to tailor colormaps to their data context. We accomplish this by treating colormap design as a path planning problem through the CIELAB colorspace with a context-specific reward model. In an evaluation with twelve scientists, we found that Cieran effectively modeled user preferences to rank colormaps and leveraged this model to create new quality designs. Our work shows the potential of active preference learning for supporting efficient visualization design optimization.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642903

CCS CONCEPTS

• **Human-centered computing** → *Interactive systems and tools*; **Visualization systems and tools**; • **Computing methodologies** → *Q-learning*; *Learning from demonstrations*; **Learning from implicit feedback**; **Active learning settings**.

KEYWORDS

visualization, colormaps, design optimization, preference learning

ACM Reference Format:
Matt-Heun Hong, Zachary Nolan Sunberg, and Danielle Albers Szafir. 2024. Cieran: Designing Sequential Colormaps via In-Situ Active Preference Learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3613904.3642903

1 INTRODUCTION

When visualizing data, analysts frequently need to decide what colors to use in a chart. We focus on the design of sequential colormaps, which must be orderable, appear smooth, include discriminable colors, and adhere to perceptual uniformity [8]. Beyond these perceptual guidelines, colormaps should also have aesthetic appeal [48], which can depend on color-data semantics (e.g., cool–warm, positive–negative affects) [4], branding, domain conventions, or personal preferences [54]. Furthermore, how we see colors applied on a chart will vary with the sizes and shapes of the colored marks [67], their spatial distribution [79], and the chart's background color [54]. These factors and their interactions make colormap design challenging for a typical data analyst.

Most often, people choose a colormap from a small gallery of options designed by experts like ColorBrewer [22] or defaults provided in visualization authoring tools, which greatly limits expressivity. Even though many other quality example colormaps are available, for example, across communities like ColourLovers,[1] the process of rendering one's visualization for each colormap across galleries and tracking best options can be time-consuming.

Furthermore, pre-existing colormaps that satisfy user needs may not exist. People can build new colormaps manually using tools like Photoshop; tools like CCC-Tool [53] aim to streamline the workflow for visualization. However, most people lack the technical or design expertise for working with color to prototype and refine colormaps effectively. This means that without tools that enable data analysts to efficiently **rank** existing colormaps or **create** new ones tailored to their data, people will continue to inadequately present data by relying on a system default [49].

We address this issue with Cieran (pronounced KEE-ruhn), an AI assistant that helps analysts rapidly rank and create sequential colormaps when designing communicative visualizations. In its training phase, Cieran adaptively and iteratively asks an analyst to choose between two different versions of their visualization each employing a different expert-designed colormap. This approach leverages prior work in color science which finds that presenting alternative choices is known to elicit the most reliable human responses for studying color preferences [54] while also being an easy task to respond to [9]. Cieran uses this choice data to update a model of aesthetic utility, which can be used to rank expert-designed sequential colormaps. It also creates new colormaps by treating their design as a path planning problem across the CIELAB space using the learned model as a reward signal [59]. All outputs monotonically decrease in L*, are perceptually uniform (Appendix A), and are interpolated using cubic B-splines [14] to meet best colormap design practices.

Contributions. Our main technical contributions (§5) include 1. an active learning strategy to induce a personalized model of aesthetic utility for colormaps from few inputs, and 2. a fast algorithm to create new colormaps given the utility model. We validate our technical approaches by conducting a user study (§6) where domain experts ($N = 12$) across physical and social sciences took part in an open-ended design task. Our findings show that Cieran can efficiently suggest and create aesthetically pleasing colormaps tailored to individual users in around two minutes of use.

The system implementation of the above approach is a publicly accessible Python package[2] for use within Jupyter Notebooks to rank and create chromatic colormaps according to a user's indicated aesthetic preferences. Cieran's workflow is as follows:

- ask the user for a required (seed) component color,
- fit expert-designed colormaps to this color and process them to be ordered, smooth, and perceptually uniform,
- iteratively display a few (< 15) pairs of expert-designed colormaps to compare on the user's target visualization,
- rank all expert-designed colormaps from the comparisons,
- create a new processed colormap that matches the user's aesthetic preferences,
- display a selection tool for the user to choose their colormap from the ranked choices.

2 BACKGROUND

Automation can support the design of effective visualizations. For example, tools like ShowMe [42] and Draco [50] provide visualization recommender systems grounded in known best practices or experimental results. Alternatively, visualization linters [11, 26, 44] assist in reviewing and identifying potentially ineffective or misleading design practices. These tools emphasize generalizable concepts of perceptual effectiveness (e.g., the ability of a population to accurately conduct visualization tasks within a controlled study [56]), while paying limited attention to residual aspects of design like personal preferences.

In contrast, the human-in-the-loop approach of ViA [24] incorporates personal preferences into visualization recommendations by asking the user to input their objectives and respond to suggestions. However, using ViA requires the user to provide significant scaffolding for the system to work by manually inputting parameters such as importance weights, tasks, spatial frequency, and domain type. By focusing on the complex space of colormap design, our work considers how visualization optimization [55] could be automated with an analyst-in-the-loop by querying them with simple questions, such as: "Which chart do you like better?" We ground our work in the science and engineering of colormap design as well as active preference learning.

[1] https://www.colourlovers.com/

[2] https://github.com/matthhong/cieran/

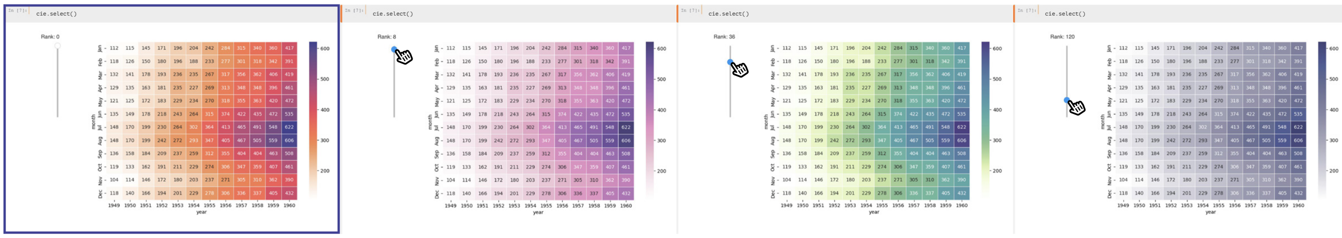

Figure 2: Cieran allows analysts to quickly design a colormap for a visualization. Cieran creates a new colormap (left) and ranks existing expert-designed colormaps (the three sorted options to the right). After training Cieran, the user makes a selection using a slider widget, with the new and the most useful example colormaps sorted to the top. This gives users the final agency over the colormap design.

2.1 Quantifying Colormap Utility

Designing useful colormaps remains a fundamental challenge in data visualization. Bujack et al. [8] provided mathematical definitions for three properties of perceptually-effective sequential colormaps: order, discriminative power, and uniformity. Discriminative power, a quality which aids performance on value comparison tasks [38], can be improved by varying a colormap's component colors across hue and/or chroma to increase total color variance.

Experts acknowledge that chromatic colormaps should also be aesthetically pleasing [48]. But the aesthetic utility of a colormap will vary with the characteristics of the audience (e.g., personal preference or past experience [2]), the specific dataset (e.g., semantics [61] and distributions [79]), and the problem space (e.g., the target visualization [67] or the data domain [13]). For example, how we see colors on a chart varies depending on the sizes, the shapes, and the spatial distribution of the colored marks [67, 79].

Consequences of these factors and their interactions are so prevalent that tools like Tableau[3] automatically modify colormaps for different visualization techniques; for example, by decreasing the color lightness values for area charts and choropleth maps [7]. However, such modifications do not take into account dimensions of aesthetics other than the chart technique used, and quantifying the aesthetic value of a colormap with more specificity remains an open problem in visualization [23, 36].

2.2 Tools for Designing Sequential Colormaps

To help domain experts customize colormaps for their target context, ColorBrewer [22], matplotlib [27] and online communities provide galleries of expert-designed colormaps. However, searching across hundreds of colormaps for the right look-and-feel can be time-consuming, and default options may not satisfy user needs.

With tools like Photoshop,[4] people can create new color gradients. Tools like ColorCAT [46] and CCC-Tool [53] seek to simplify both colormap creation and validation by allowing users to manually input and adjust control points and providing metrics for assessing the output colormap's perceptual utility. Using such tools requires substantial manual effort and design expertise. Therefore, AI-assisted tools have leveraged existing examples to guide colormap construction. For example, deep learning can enable colormap style transfer from existing images to new charts [37, 64, 76, 77]. However, users may not have quality example images.

Instead, systems can directly leverage the hundreds of expert-designed colormaps as examples. ColorCrafter [62] first mines a corpus of 222 colormaps for their structural features and outputs nine model curves that are characteristic of common design practices. People can apply a seed color to one of the model curves and customize it using affine transforms. ColorMoves [60] allows users to mix-and-match segments of example colormaps. While such a paradigm may be expressive, the tool entrusts users with the combinatorial task of mixing and matching colormap segments.

In our work, we aim to build a tool that automatically suggests and creates colormaps from examples, in a manner that is driven by users' aesthetic preferences. We hypothesize that active preference learning can be used to rank colormaps according to aesthetic utility and automate the assembly of new colormap structures.

2.3 Preference-Based Design Optimization

Prior work in computational design have applied active preference learning to enhance the quality of images and renderings. Their technical contributions often involve algorithmic improvements to Bayesian Optimization from discrete choice data [6], including one-to-many comparisons or learning directly from slider interactions [34] or brush strokes [33]. Preference learning leverages the power of alternative choice tasks to collect reliable information about the utility [30] of a design artifact using a simple input modality.

Pairwise comparison also elicits the most reliable human responses for studying color preferences [54]. Following prior work, BayesOpt could also be used to optimize certain types of colormaps including Cubehelix [18], which parameterizes colormap structures based on a start color, saturation levels, and emphasis on different regions of interest. However, Cubehelix colormaps have a distinct spiral structure that constrains the space of designs.

We instead model the aesthetic utility of non-parametric sequential colormaps using a preference-based reward learning (PbRL) algorithm [12]. PbRL algorithms are typically used to steer the behavior of virtual and physical robots to complement inverse reinforcement learning [1, 32] or reinforcement learning from human feedback [3, 69]. We hypothesize they can also guide the design of colormaps, which are continuous and smooth trajectories in a 3D space that a steerable robot might traverse.

Cieran uses this intuition to model the aesthetic utility of colormap curves from a few pairwise comparisons to potentially capture a wide range of aesthetic considerations, and in turn, optimize the design of sequential colormaps.

[3]https://www.tableau.com/
[4]https://www.adobe.com/products/photoshop/

3 SYSTEM OBJECTIVES

Quantifying the aesthetic value of colormaps remains an open problem in visualization [23, 36]. Factors that are difficult to generalize such as personal preferences, data semantics, and data patterns can influence the aesthetic appeal of colormaps. Although galleries like ColorBrewer, matplotlib's colormap library, and online communities like ColourLovers allow users to explore a range of expert-designed options, manually sifting through colormaps takes significant trial and error. Furthermore, default options limit expressivity. Tools like Photoshop or CCC-Tool can support custom colormap creation, but they require considerable manual tuning, along with the time and expertise to effectively navigate this process. Our goal is to simplify the process of customizing sequential colormaps by reducing the need for user expertise and manual cost.

Our approach builds on the work by Smart et al. [62], which mined structural patterns across existing expert-designed colormap curves. By incorporating their colormap corpus within a data analysis software, we aim to enable people to **efficiently rank many example colormaps in the context of the data visualization being designed**, and in the process learn a model of aesthetic utility for colormaps within the user-specific context. We then aim to use both the learned utility model and the expert examples to **create new aesthetically pleasing colormaps** to suggest as additional options. Recent enhancements to CCC-Tool [52] include post-processing steps to ensure colormaps meet mathematical formulations of perceptual guidelines outlined by Bujack et al. [8]. Their work suggests that tools to create sequential colormaps can and should **automatically enforce perceptual guidelines** with either constraints or post-processing algorithms.

In summary, we developed a colormap design tool that accomplishes the following objectives:

- **O1**. Allow people to efficiently and effectively sort expert-designed colormaps according to their look-and-feel on the target visualization and given their specific data context.
- **O2**. Create novel colormaps, also driven by expert examples, that provide scenario-specific alternatives to existing options.
- **O3**. Ensure that each colormap satisfies perceptual guidelines for colormap design, such as linear order, smoothness, and perceptual uniformity.

4 CIERAN: USER INTERFACE

Cieran is an open-source Python package that interfaces with a user through a Jupyter Widget. This paradigm allows data analysts to both stay within their typical workflow and to design colormaps directly on their target visualization. Therefore, Cieran can be used to rapidly enhance the quality of any colormapped visualization that can be rendered inside a Jupyter Notebook *in situ* (**O1**).

Using Cieran involves three steps. First, the user **a. initializes** a Cieran object with a target visualization and color. The system then displays a widget interface where the user **b. teaches** a model their aesthetic preferences for colormaps through pairwise choices (Figure 1). Based on these inputs, Cieran allows the user to **c. select** a colormap from a list of options sorted (Figure 2) according to their aesthetic utility.

a. Initialization. Users initialize Cieran with a callback function that accepts a `matplotlib.colors.ListedColormap` object and displays the target colormapped visualization, in other words, `Cieran(draw: Callable[[ListedColormap], None])`. Then, the user specifies a seed color that they wish to include across all options with the `set_color(color: str)` instance method. The seed color provides an initial entry for the user to indicate their preferences. It also both reduces the number of example colormaps and constrains the search space of new colormaps.

b. Training. Cieran is trained by the user iteratively indicating their preference across pairs of example colormaps (Figure 1). This training interface is displayed by invoking the `teach(N: int)` instance method. By default, Cieran queries for $N = 15$ text responses, with 1 indicating preference for the left colormap option, 2 indicating right, and 0 indicating indifference. Pairwise comparison is known to elicit the most efficient and reliable responses for modeling color preferences, compared to rank ordering or absolute value judgments (e.g., a Likert scale response) [54].

c. Outputs. The trained utility model is used to automatically accomplish two design objectives: ranking all example colormap structures (**O1**) and creating a new colormap (**O2**) by piecewise combining example colormaps. Afterwards, the user invokes a slider widget with the `select()` instance method, where the new and the most useful example colormaps are sorted to the top of a vertical slider (Figure 2). The colormap selected with the slider can be accessed through the `cmap` instance property. People can use the `ListedColormap` object interface to further refine the selected colormap (e.g., to resample a smaller number of discrete colors or to represent it as an array of hex values for later re-use).

All colormaps displayed to the user vary monotonically in lightness, and consist of 256 colors sampled at ΔE_{2000}-equidistant rectified arc lengths along an approximate cubic B-spline curve [14] to maintain perceptual uniformity and smoothness (**O3**, Appendix A).

5 CIERAN: TECHNICAL DETAILS

Since colormaps are three-dimensional curves across a colorspace, the problem of finding an aesthetically pleasing colormap is akin to finding a high-utility path within an environment. Cieran executes on this idea by formulating a search space of colormap trajectories, learning a utility model for trajectories from user preferences, and using this model to both rank example colormaps and search for new paths through a colorspace (Figure 3).

5.1 Technical Overview

Colorspace Quantization. Prior to the interpolation of a colormap curve, we define a colormap *trajectory* $\Upsilon = (s_0, a_0, \dots)$ as a sequence of control point colors in the CIELAB colorspace [41] traversed by a virtual agent based on actions taken across a graph-based environment (S, A) where:

- S, the state space, is a set of 512 unique CIELAB colors, and
- A_s, the action space of a color $s \in S$, contains possible movements towards neighboring colors.

This environment is inherently acyclic (i.e., a DAG), since the agent cannot move backwards in lightness values in order to preserve lightness monotonicity (**O3**). Section 5.2 provides details about the construction of this environment. Our methodology of

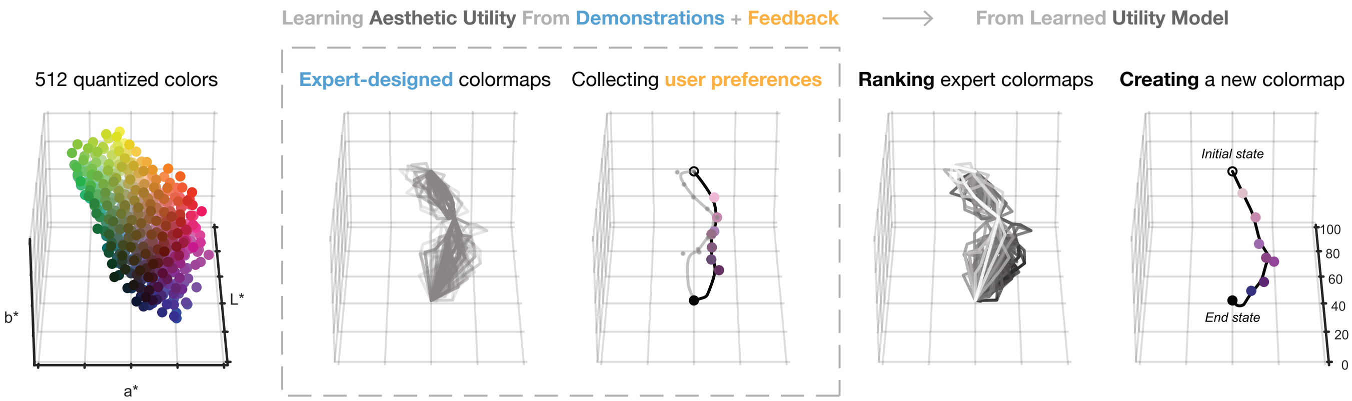

Figure 3: Cieran is a path planning agent that helps rank and create sequential colormaps with a human in the loop. Cieran first constructs a graphical environment of possible colormap trajectories through a user-specified seed color based on expert demonstrations of colormaps (second subfigure). It then induces a model of aesthetic utility given a small number of pairwise comparison data (the third subfigure highlights the user's selection) collected through a Jupyter Widget, which adaptively and iteratively presents pairs of best candidate colormap examples applied to the user's target dataset. Using the learned utility model, Cieran will score all expert-designed colormaps (fourth subfigure), and create a new quality colormap via path planning (fifth subfigure).

selecting these 512 approximately equidistant colors in the CIELAB color gamut leads to the mean Euclidean distance between possible control point colors to around $\Delta E = 5$, which is the just-noticeable color difference threshold for untrained observers [47].

Learning to Rank. Cieran's training phase learns to score each trajectory Υ with a measure of its aesthetic utility—a user- and context-specific reward function $R(\Upsilon)$—by leveraging a user's pairwise comparison responses. Assuming perfectly accurate comparisons, precisely ranking n example colormaps would require the user to make $O(n \log n)$ comparisons in the worst case. Our active learning paradigm adaptively elicits responses to the most informative query pairs to approximate R efficiently from noisy human comparison data. Section 5.3 provides details about this active learning strategy that can learn R from 15 human labels.

Searching for a Novel Colormap. Once $R(\Upsilon)$ is learned, we could, in principle, use it to score every trajectory in the DAG to find the best novel trajectory (i.e., a combination of piecewise segments from existing colormaps). However, an algorithm that enumerates and scores all possible combinations of S and A would have exponential time complexity [68]. Given a graph, dynamic programming algorithms can achieve path planning to find only the optimal path. However, we define the aesthetic utility of a colormap as a holistic quality (i.e., the harmony of each color to all other colors), not as the sum of utilities across locally adjacent color pairs; such non-additive cost functions preclude the use of combinatorial optimization. Section 5.4 details an approximate dynamic programming algorithm that can still efficiently search for novel colormaps with high cumulative rewards in our graph-based environment.

Cieran forces all colormap trajectories to start from white and end at black to (a) address the potential need for using white to represent zero values and (b) provide matching terminal states across trajectories to simplify the search algorithm. In practice, Cieran's colormap curves (interpolated from the trajectories) are truncated at $L* = 10$ by default to exclude the black colors, in accordance with best practices [78].

5.2 Colormap Quantization

We first define (S, A), the graph-based environment from which Cieran will sample candidate trajectories through the CIELAB colorspace (Figure 3, first subfigure). This DAG comprises quantized structures from a corpus of 222 expert-designed colormaps from commercial platforms (e.g., Tableau and R) and communities of practice (e.g., ColourLovers) collated in prior work ColorCrafter [62]. Using this corpus constrains the state and action spaces to those reflecting expert designer practices, and quantizing each colormap allows Cieran to create novel colormaps by joining piecewise segments of existing colormaps.

As in ColorCrafter, we first sample nine equidistant points from each expert-designed colormap. We then align each colormap to a user's seed color by first finding its closest corresponding color with respect to L^*. Then, we generate two alternative options for each of the nine-color ramps. For the first alternative, we rotate the colormap in the hue plane (i.e., the a^*–b^* plane) by the angular difference between the seed and corresponding color. We then translate the colormap in the L^*–C^* plane to complete the alignment. For the second alternative, only the displacement across the L^*–C^* plane between the two colors is used to translate the colormap.

For the state space S, Cieran first quantizes CIELAB by generating an approximately equidistant set of 512 colors within the gamut using Halton sampling [20], a quasi-random process commonly used in robotics to generate evenly-spaced samples from configuration spaces [16]. To ensure even dispersion, these positional samples are then passed through the Lloyd-Max algorithm [39, 43] until convergence. For the action space A, the nine colors in each new colormap are matched to their nearest colors in the quantized color space. If a pair of colors (s, s') are adjacent in any quantized colormap, we add an action s' to A_s. Finally, every lightest color in each colormap becomes the action space of the white state $(100, 0, 0)$, and every darkest color has an action that corresponds to the black state $(0, 0, 0)$.

Table 1: Reward Features and Values

Feature	Description	Value
$\{k_1, \ldots, k_8\}$	perimeter distance	$\mathbf{k} \cdot \theta_{1:8}$ when $s' =$ black
ℓ	landing reward	10 when $s' =$ black
m	chroma inclination	$m * \theta_m$ when $s' =$ black
n	moving penalty	-0.01 per action

5.3 Ranking Expert-Designed Colormaps

For each trajectory Υ through the DAG, we can define a feature vector Φ that describes its characteristics. Then R, the utility of a trajectory, can be defined as a linear combination these features, such that:

$$R(\Upsilon) = \theta \cdot \Phi(\Upsilon) \tag{1}$$

Algorithm 1 and Figure 5 describe the strategy to actively learn θ from pairwise colormap preferences, which requires the specification of reward features, a belief model, and an acquisition model.

5.3.1 Reward Model. Our reward model considers that (1) curves should go through preferred colors (e.g., red) or color families (e.g., warm colors) that a user finds aesthetically pleasing, (2) colormaps should be long enough to be discriminable, but not too long to introduce artifacts like hue banding [5] and (3) that lightness and chroma characteristics are key factors for effective sequential colormaps [8]. The reward model has nine path features $\{k_1, \ldots, k_8\}$ and m (Table 1) with corresponding unknown weights $\theta_i \in [-1, 1]$, the preference model learned from user choices.

Perimeter distances k are the normalized shortest distances between the curve and eight corner points on the boundaries of CIELAB (Figure 4). These distances both reflect preferences for certain color categories and are also indicators of a curve's global length. Increased curve length may improve the resulting colormap's discriminability [8]; however, highly saturate but less preferred hues (i.e., closer proximity to the corresponding perimeter points) are likely to lead to lower aesthetic utility [17]. A negative weight θ_k indicates that the user prefers a colormap that is close to the color family corresponding to the perimeter k.

We model preferences for color chroma (akin to saturation) characteristics as the slope m of each colormap trajectory across the $L^* - C^*$ plane computed using linear regression. We normalize the slope such that the colormap with the largest absolute slope has a value of $|m| = 1$. A negative weight θ_m leads to the lighter colors being relatively more saturated.

5.3.2 Belief Model. We initialize the preference model with a non-informative prior by uniformly sampling points from a unit hypersphere. Given the conditional independence assumption, we can iteratively update the model as:

$$P_{t+1}(\theta) \propto P(y_t \mid x_t, \theta) P_t(\theta) \tag{2}$$

where $x_t = (\Upsilon_t^1, \Upsilon_t^2)$ is the pairwise preference query and $y_t = k$ is the user's response to the query with $k \in \{1, 2, \varnothing\}$, where 1 indicates a preference for the option on the left, 2 on the right, and $\varnothing$ indicating indifference.

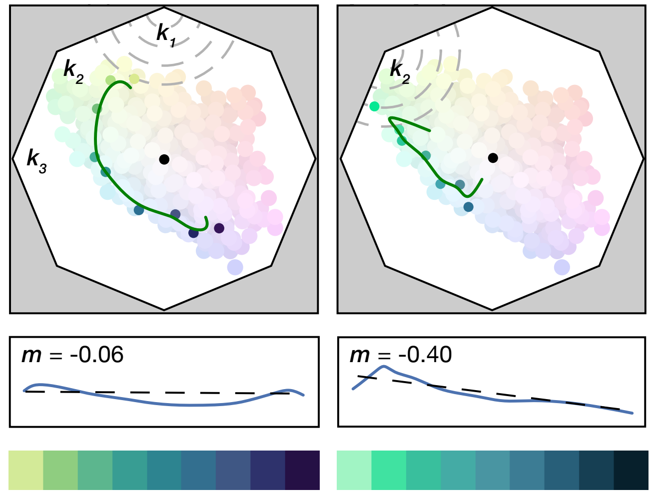

Figure 4: A visual explanation of the features and weights θ (Table 1). $\{k_1, \ldots, k_8\}$ describes the shortest distance from each of the perimeters in the a^*–b^* plane to the colromap. m is the slope of the colormap in the L^*–C^* plane. These two colormaps represent two trajectories going through the same seed color #186E8D. The colormap on the left is recommended by Cieran when setting $\theta_1, \theta_2, \theta_3$ to -0.5, indicating a preference for a colormap that goes through greenish yellow tones. The colormap on the right is recommended when setting $\theta_2 = 0.5$, indicating a preference for a colormap that only miminizes distance to the green perimeter of the gamut, while also setting $\theta_m = -0.5$, indicating a preference for a colormap that is highly saturated at lower $L*$ values.

To model a noisy yet rational human expert selecting k, we utilize the softmax likelihood function:

$$P(y_t = k \mid x_t, \theta) = \frac{\exp(R(\Upsilon_t^k))}{\sum_{\Upsilon \in x_t} \exp(R(\Upsilon))}. \tag{3}$$

Some colormap pairs may be equally preferable. We therefore include an "indifference" response [9] by introducing the *minimum perceivable difference threshold* $\delta \geq 0$ [35] such that:

$$P(y_t = \varnothing \mid x_t, \theta) = (\exp(2\delta) - 1) \prod_k P(y_t = k \mid x_t, \theta) \tag{4}$$

where:

$$P(y_t = k \mid x_t, \theta) = (1 + \exp(\delta + R(\Upsilon^{k'}) - R(\Upsilon^k)))^{-1} \tag{5}$$

in which Υ^k and $\Upsilon^{k'}$ are the two colormaps being compared and $k \neq k'$. Equation 5 reduces to Equation 3 when $\delta = 0$. Prior work shows that the choice of δ introduces minimal variability, so our algorithm relies on a standard value of $\delta = 0.01$ [10]. Given the above definitions, the Metropolis-Hastings algorithm [45] can adaptively generate samples from the posterior distribution over θ after obtaining each participant response k.

5.3.3 Acquisition Model. To sample a pair of colormaps, we use the query-by-disagreement method from Katz et al. [31]. This technique can sample a pair of colormap trajectories that maximize both their potential for aesthetic utility as well as disagreement (i.e., likely differences between colormap utilities).

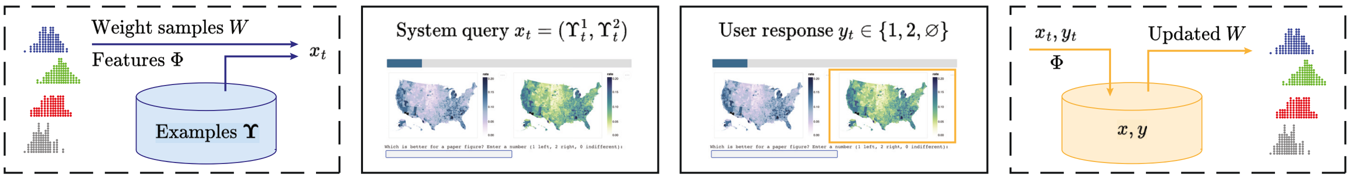

Figure 5: Overview of the active learning-to-rank process (Algorithm 1). [Left] At each iteration, samples W are first drawn from current $P_t(\theta)$ using Metropolis-Hastings. Next, these samples and the feature function Φ are used to acquire an informative query pair from the example colormap corpus (Eq. 6). [Middle] The user observes this query and makes a pairwise value judgment. [Right] User responses are used to update $P_t(\theta)$ using with the feature function Φ once again (Eqs. 1 through 5), and the iterative process continues.

Let W be samples from the posterior distribution over θ, and let W_i and W_j be the ith and jth samples of W. The optimization problem for generating the next query becomes:

$$\text{argmax}_{i,j}\, P(\theta = W_i)P(\theta = W_j) + \lambda \left\| W_i - W_j \right\|_2 \tag{6}$$

where $i \neq j$. We can estimate $P(\theta = W_i)$ using Gaussian KDE. The sampled query pair x will be colormaps whose trajectories (Υ^1, Υ^2) maximize $W_i \cdot \Phi(\Upsilon)$ and $W_j \cdot \Phi(\Upsilon)$ respectively. $\lambda \geq 0$ is a temperature parameter to incentivize divergent queries; we set it to 500 based on piloting and recommendations from Katz et al [31].

Algorithm 1 Learning θ From Colormap Preferences

Input: Υ, Φ, L ▷ Trajectory set, feature vector, iteration limit
Output: $\overline{W}$ ▷ Parameter estimates θ

```
t ← 0, x ← ∅, y ← ∅, W ← MetropolisHastings(x, y, 0)
repeat
    (Υ¹, Υ²) ← QueryByDisagreement(Υ, W, Φ)
    x ← x ∪ (Υ¹, Υ²)
    y ← y ∪ ObtainUserPreference(Υ¹, Υ²)
    W ← MetropolisHastings(x, y, W̄)
    t ← t + 1
until t = L
```

5.4 Searching for Novel Colormaps

5.4.1 General Q-Learning Approach. Cieran utilizes an approximate dynamic programming algorithm known as Q-learning [74], as outlined in Algorithm 2. The algorithm observes a reward r after taking each action and estimates Q—the value of the action (i.e., appending a color to the current colormap trajectory)—as:

$$Q(s,a) \leftarrow (1-\alpha)\, Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') \right). \tag{7}$$

Q-learning is typically used to solve a Markov decision process (MDP) where r is defined only in terms of the states and actions, and the utility of a trajectory is the sum of the rewards at each step. However, Cieran evaluates colormaps based on their holistic aesthetic utility, and not as a sum of utilities for adjacent pairs of colors. Therefore, its agent will only receive positive rewards at the last step based on the entire trajectory taken.

Beginning its search from the color white, at each iteration t, the algorithm selects an action using an epsilon-greedy strategy and observes a reward. As described in Table 1, there is a default cost of $n = -0.01$ to every action to prevent unnecessary movements which do not add to the cumulative reward. This action cost discourages control colors that cause unnecessary chroma or hue variance without contributing to the colormap's aesthetics.

Based on chosen actions and observed rewards, Cieran iteratively updates the state-action value function $Q(s,a)$ using the incremental update rule (Equation 7) until it reaches the black (end) state. If the next state is the black state, a landing reward of $\ell = 10.0$ is added to the cumulative reward along with $R(\Upsilon) = \theta^\top \Phi(\Upsilon)$ based on the features of the final colormap trajectory.

When Cieran reaches the black (end) state, if the completed trajectory Υ meets the following criteria, it is saved as the new Υ^*:

(1) it is novel, as in not already in the corpus (since we have already scored all existing colormaps);
(2) it passes through at least two points between white and black, one of which is the seed color;
(3) every point along its interpolated path is within the gamut;
(4) and it has the highest cumulative reward (i.e., the highest aesthetic utility).

5.4.2 Hyperparameters. Our Q-learning environment uses a non-deterministic transition model where $T(s,a,s') = 0.95$ when $a = s'$ but transitioning to a randomly different adjacent state with probability 0.05. Since our environment is not an MDP typically suited to Q-learning, incorporating additional noise to the reward signal makes the algorithm more robust. Q-learning requires a discount parameter γ, but because every trajectory has the same absorbing state (black), $\gamma = 1.0$ [75]. Both the exploration parameter and learning rate are set to $\epsilon = 0.1$ and $\alpha = 0.1$ based on piloting.

We primarily follow standard Q-learning practices except for the initialization of Q. Specifically, the initial action-value estimates Q_0 are set to 100 for all actions, which is significantly higher than what can be observed from the reward model. This Optimistic Q-learning [15, 66] approach encourages early exploration by the model by making all unexplored options appear better than they actually are. We evaluate the efficacy of this approach in Section 6.3.

5.5 Implementation

We implement Algorithm 1 with modifications to the `APReL` package [10], which uses `numpy` [21] and `scipy` [72]. We implement Algorithm 2 with `numpy` and `networkx` [19]. B-spline interpolation of curves and sampling ΔE_{2000}-equidistant colors utilize `coloraide` [51]. Each colormap is a `matplotlib.ListedColormap` [27] class instance. We implement the interface using `ipywidgets` [29].

Algorithm 2 Searching for a Novel Colormap From θ

Input: $\theta, \Phi, L, \epsilon, \alpha, Q_0$ ▷ Preference model, iteration limit, and hyperparameters
Output: Υ^* ▷ Found high-utility trajectory
for $\forall s, a$ **do**
    $Q(s,a) \leftarrow Q_0$ ▷ Initial optimistic estimate
end for
$t \leftarrow 0, \Upsilon^* \leftarrow \emptyset, r^* \leftarrow -\infty$
repeat
    $s \leftarrow white, \Upsilon \leftarrow \{s\}, r \leftarrow 0$
    repeat
        $a, s' \leftarrow \textsc{EpsilonGreedy}(\epsilon, Q, s)$
        $\Upsilon \leftarrow \Upsilon \cup \{a, s'\}$
        $r \leftarrow \textsc{ComputeReward}(\Upsilon, \theta, \Phi)$
        $Q(s,a) \leftarrow Q(s,a) + \alpha\,[r + \max_{a'} Q(s', a') - Q(s,a)]$
        $s \leftarrow s'$
    until $s = black$
    if $r > r^*$ **then**
        $\Upsilon^*, r^* \leftarrow \textsc{UpdateResult}(\Upsilon, r)$
    end if
    $t \leftarrow t + 1$
until $t = L$

6 SYSTEM EVALUATION

We evaluated whether Cieran can effectively rank and create colormaps that satisfy analysts' design needs given a small set of preference inputs. The study first involved a holistic mixed-methods evaluation of the system with fourteen participants who each designed a set of colormaps using Cieran. We then evaluated our Q-learning implementation (Algorithm 2) using participants' learned preference models.

6.1 Study Protocol

6.1.1 Participants. Fourteen researchers participated in the study. The first two participants (E1 and E2) were experts in using colormaps in visualization and served as pilot participants. Although we do not include their results in our quantitative analysis, our discussions with them informed our study design, and their feedback is included in the qualitative analysis.

We recruited 12 additional social or physical scientists (P1-P12) from the United States who have all had experience building visualizations in a scripting language such as Python, R, JavaScript, or SPSS. All participants were native or fluent English speakers over 21 years old. No participant self-reported color vision deficiency, although this was not part of our exclusion criteria.

6.1.2 Stimuli. Participants designed colormaps for a discretely-binned 2D kernel density plot rendered in Seaborn [73]. This plot was accompanied by black discrete histograms of the corresponding marginal distributions to the right (y distribution) and top (x distribution) of the heatmap (Figure 6). We chose 2D kernel density plots so that the colormaps would be mapped onto smooth distributions and display multiple distinct layers of color values across the length of the colormap, while also minimizing potentially distracting structural artifacts. The histograms allowed participants to verify that every colormap represented the same dataset.

Each participant saw one of three pre-generated datasets. The datasets were isotropic Gaussian blobs in 2D space generated using `sklearn` designed to ensure that the spatial distribution of data covered most of the display area. These stimuli also avoided symbolic shapes and hotspots that would pop-out and distract the participant from the global color characteristics. We randomly assigned one-third of participants to each dataset condition.

6.1.3 Procedure. We conducted a mixed-methods study using a web interface in a hybrid remote/in-person setting. All verbal and physical interactions were recorded through Zoom screenshare.

We first collected informed consent and basic demographic information. We then asked participants a series of questions about their experiences with visualization and color, including their favorite color. We then introduced the heatmap, describing it as a visualization that allows us "to aggregate [scatterplot] points, making the distributions easier to see. As it turns out, heatmaps can come in a variety of different colors." They were instructed to make a sequence of preferential choices between variants of colormaps mapped onto the heatmap to design a journal paper figure.

Before the formal trials, the participant and the interviewer discussed how the participant would usually go about designing a colormap in the given scenario to elicit baseline strategies for colormap design. Participants then moved on to the study, where we collected both quantitative and qualitative user data.

6.1.4 Quantitative data collection: Participants made 15 comparisons over three phases (one phase per user-specified seed color), totaling 45 comparisons per participant. We observed during pilot studies that people tended to lose engagement after 15 comparisons. The last task in each phase was manually ranking colormaps, and the whole phase was designed to last at most two minutes.

To reduce potential confounds from color pickers, participants specified a seed color from the Tableau10 color palette [65], an expert-designed palette of ten discrete colors designed for categorical visualization. The first phase used the color in Tableau10 that was closest to the participant's favorite color, but participants were free to choose any color, including a previously used color, in other phases. For each pairwise comparison, the participant could indicate their preference for left (1), right (2), or indifferent (0) input using a textbox. They were instructed to choose 1 or 2 if there was even the slightest preference for one of the colormaps or if they simply disliked one more than the other.

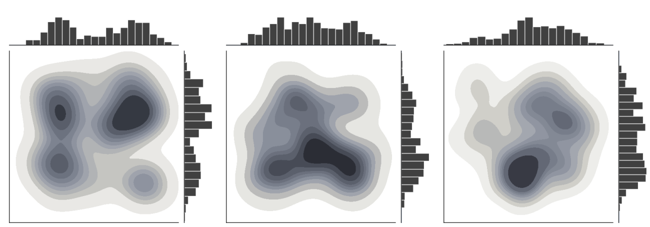

Figure 6: The three dataset stimuli used in our study. Participants designed colormaps for a discretely-binned 2D kernel density plot rendered in Seaborn [73]. This plot was accompanied by black discrete histograms of the corresponding marginal distributions to the right (y distribution) and top (x distribution) of the heatmap.

Following each set of 15 comparisons, in place of using the slider selection tool (Figure 2), the participant rank-voted four different colormaps: (1) the novel colormap found through search, (2) the top-ranked expert-designed colormap according to its learned aesthetic utility, (3) the median-ranked expert-designed colormap, and (4) the lowest-ranked expert-designed colormap. The four choices were laid out in a 2x2 grid in a random order. Participants input their votes (1–4, with 1 indicating the highest aesthetic utility, and 4 indicating the lowest) using an integer widget.

6.1.5 Qualitative data collection: Participants were encouraged to think aloud and verbally reflect upon their choices at any point during the study. The experimenter also asked questions during the study, including why the participant made a given preference choice in the first trial of the first block and when participants made an unexpected choice based on their past behavior and available information. After the study ended, the participants were asked whether they had any other thoughts during the study before being briefed on the study objective, which was to evaluate the efficacy of a colormap recommendation tool.

6.1.6 Analysis. We coded the audio transcripts of the interview for insights into the participants' design objectives, prior experiences with colormap customization, and their experience using Cieran. The ranked-choice data from the quantitative data collection was analyzed using the Plackett-Luce model [40] to compare the performance of the four colormap options (new, top-rank, median-rank, and last-rank).

6.2 Results

6.2.1 Quantitative Results. We report inferential statistics, means, and standard errors (means ± standard errors) for the differences in ranked-choice votes (or, worth[5]) across Cieran's new and example colormaps (Figure 7).

We did not find a statistically significant difference between new colormaps and the top-ranked expert-designed colormap ($\mu = 0.31 \pm 0.30, z = 1.05, p = 0.30$). However, new colormaps were preferred to the median-ranked ($\mu = -0.67 \pm 0.33, z = -2.02, p < 0.05$) and the lowest-ranked ($\mu = -2.75 \pm 0.50, z = -5.42, p < 0.001$) expert-designed colormaps. Similarly, the top-ranked expert-designed colormaps were preferred over the median-ranked ($\mu = -0.98 \pm 0.33, z = -2.96, p < 0.01$) and the lowest-ranked ($\mu = -3.06 \pm 0.51, z = -5.98, p < 0.001$) expert-designed colormaps.

These results suggest that:

(1) Cieran effectively ranks colormaps according to aesthetic utility from a small set of personal preference data, and
(2) In aggregate, people found both new and higher-ranked colormaps to be useful, but in some instances (12/36) people may find the new colormap preferable to all expert-designed options, including the top-ranked.

6.2.2 Qualitative Results. Our qualitative results reinforced the idea that the aesthetic value of colors is a concept that is difficult to generalize across a population. Many people indicated an aversion to specific colors, such as yellow (P1, P3, P4), green (P1, P3), pink (P1), and brown (P9). Some disliked high-chroma colors, such as "electric" blues, purples, and greens (E2, P1, P8), indicating that they "hurt [their] eyes" (P8). However, some specifically preferred yellow (P5), green (P5), "electric" blue (P6), purple (P7), and pink (P8).

Cieran enabled participants to explore a wide variety of colormaps. Many participants ($n = 7$) favored "vibrant" (P3) or multi-hue colormaps (E2, P5) and indicated that high hue variance made data more legible (E2, P2, P3, P5). However, such preferences were not universal: many other participants ($n = 7$) avoided high variance in hue, worrying that this could distract from communicating data clearly (E1, P1, P10, P2, P4, P12) and might "overwhelm" people (P12). Many participants avoided color sequences with highly saturated colors, regardless of whether the colormaps were single-hue (P1, E1, P10) or multi-hue (P7, P8).

Participants who preferred multi-hue colormaps talked about color harmony (E2, P8, P9), color combinations (P5), or how colors "work together" (P7) to describe their design objectives. These participants talked about colormap utility in terms of whether they were "distinguishable" (P1, P9), "functional" (P1), or "crisp" (P3); or, in contrast, whether they were "blurry" or "mushy" (P3). They described perceived high-utility colormaps as possessing qualities of

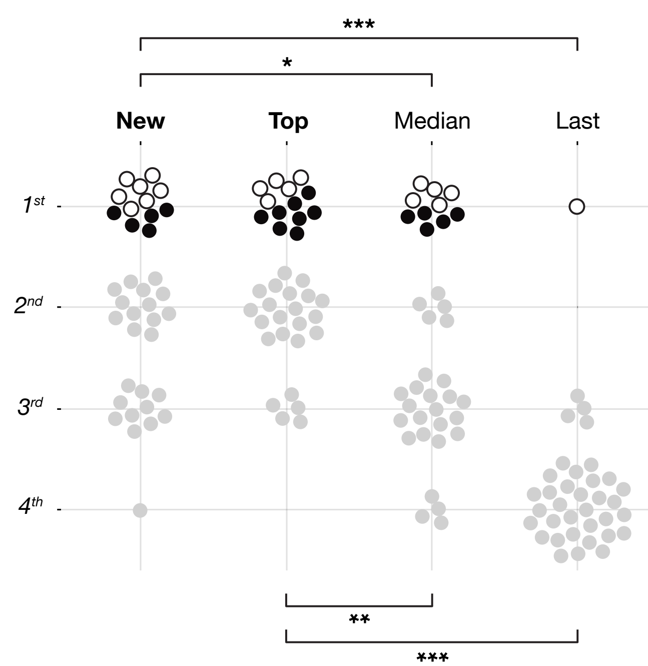

Figure 7: Ranked choice votes from the user study. The final task in each phase was to manually rank a shuffled set of four colormaps: Cieran's new colormap and the top-, median-, and last-rank expert colormaps (according to the learned preference model). The new colormap and the top-ranked expert colormap were manually placed higher than others, and in both cases, the differences were significant (* < .05, ** < .01, *** < .001). Of the 36 1st place votes across the top row, filled circles are votes from participants optimizing for colormaps with low hue variance; unfilled circles are votes from participants optimizing for multi-hue colormaps.

[5] https://cran.rstudio.com/web/packages/PlackettLuce/vignettes/Overview.html

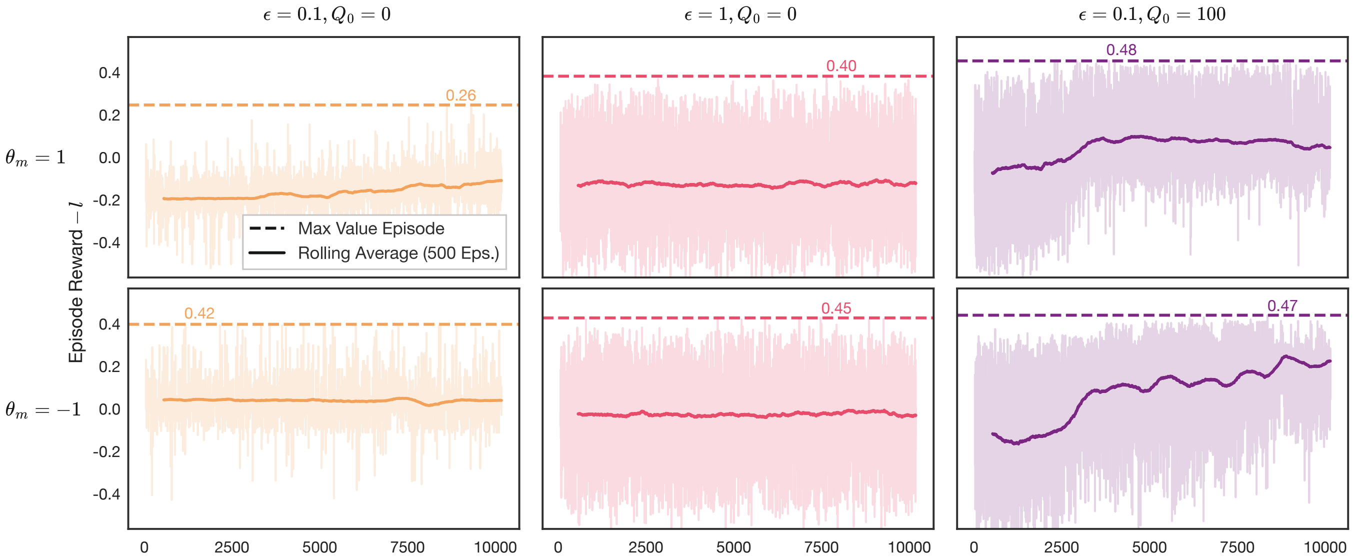

Figure 8: Optimistic Q-learning as an adaptive path search algorithm across a DAG. These cumulative reward plots compare the behaviors of three algorithms by depicting the total episode reward (minus landing reward l) across 10,000 learning episodes. Each plot depicts a simulation performed with a set of hyperparameters ϵ and Q_0 and with all preference model weights θ set to 0 except $\theta_m = 1$ or $\theta_m = -1$ (see Figure 4 for explanation). Traditional Q-learning (left) with $\epsilon = 0.1$ fails to find a high-utility colormap across the DAG, performing significantly worse than random path draws ($\epsilon = 1$, center). However, Optimistic Q-learning (right) with $\epsilon = 0.1$ and $Q_0 = 100$ performs well as a search algorithm by adaptively learning to avoid actions that lead to low-utility colormaps.

"legibility" (E1, P7) or "visibility" (P9). Those who preferred single-hue colormaps seemed to operate with a different objective, often articulating this goal as "simplicity" (E1, P10, P2, P4, P12).

The only thing people seemed to agree on was that colors representing lower values should be visible against the white background (E1-E2, P1-P12), so all participants avoided colormaps with minimal variance in chroma. However, Cieran's ranking and creation capabilities were apparently robust to the variety across participants' design objectives (Figure 7). One participant noted that their preferences may vary with their mood (P5), while another noted that they were "warming up" to certain colors during the study (E2).

6.3 Post-Study Analysis of Algorithm 2

Algorithm 2 incorporates a well-known reinforcement learning algorithm to search for a high utility path across the colorspace structured as a directed acyclic graph. Since the utility of a colormap depends on its entire trajectory, the RL agent receives rewards based on actions taken multiple steps prior; such an environment is not an MDP, and the convergence guarantees typically available for Q-learning [28] do not apply. However, this RL agent's goal is not to learn a generalizable model; we incorporated the algorithm within an adaptive search procedure to simply find a useful and novel colormap (Algorithm 2, line 15) within a reasonable time limit L (which was set to 10,000 trajectory samples for the user study). In this section, we analyze our agent's behavior to validate that it still improves with experience to better guide its search.

To study the search algorithm and effects of its hyperparameters, we re-ran the Optimistic Q-learning algorithm (with $Q_0 = 100.0, \epsilon = 0.1$) for each participant 10 times using their chosen seed color and the corresponding learned preference model θ^{user}, storing the total reward of the best colormap found after each search. From this total reward we subtracted the landing reward l, which is a constant value set at 10 regardless of the user preference model or the algorithm. We then repeated this for two additional sets of hyperparameters: $Q_0 = 0.0, \epsilon = 1.0$, resulting in a random sampling strategy, and $Q_0 = 0.0, \epsilon = 0.1$, the traditional Q-learning paradigm. Finally, we employed a linear mixed effects model to quantify the effects of using the three algorithms for colormap search, accounting for the repeated measures on individual models θ^{user}:

$$\text{maxTotalReward} \sim \text{algorithm} + (1 \mid \theta^{user})$$

Our findings are summarized in Table 2. On average, the 10,000 iterations of the sampling algorithms took 4.02 seconds on a 2019 MacBook Pro and Python v3.10.12. Relative to the baseline, traditional Q-learning significantly underperformed ($p < 0.001$) in terms of its output colormap's utility. However, Cieran's optimistic approach significantly outperformed ($p < 0.001$) other approaches.

Figure 8 illustrates typical learning behaviors of the path planning agent under the three algorithm conditions. In the traditional Q-learning paradigm, the agent has a difficult time exploring efficiently within the time limit; the random sampling agent is more likely to have a serendipitous run-in with a high-utility colormap, even though the random sampling agent does not learn and adapt. On the other hand, Optimistic Q-learning quickly learns to rule out actions that lead to low-utility colormaps by exploring actions more evenly during its early learning stages. Once the agent's Q-values are updated to more realistic values, it is able to focus on color pairs with higher potential in its search for useful colormaps.

Table 2: Summary of a linear mixed effects model comparing the performance of Q-learning agents to a random sampling baseline. Estimates represent the relative difference from baseline in the cumulative rewards of output colormaps.

	Description	Estimate	Std. Err.	z	$P > \|z\|$	[0.025, 0.975]
baseline[$Q_0 = 0.0, \epsilon = 1.0$]	Random Sampling	0.550	0.115	4.789	$< 0.001^{***}$	[0.325, 0.774]
condition[$Q_0 = 0.0, \epsilon = 0.1$]	Q-learning	-0.079	0.003	-30.700	$< 0.001^{***}$	[-0.084, -0.074]
condition[$Q_0 = 100.0, \epsilon = 0.1$]	Optimistic Q-learning	**0.015**	0.003	5.920	$< 0.001^{***}$	[0.010, 0.020]

7 DISCUSSION

We demonstrated Cieran's capability to effectively rank and create colormaps from user preferences when designing visualizations. Our system addresses colormap optimization for both aesthetic and perceptual utility by leveraging a corpus of expert-designed curves and processing all output colormaps. While the prior section summarized these findings given our problem space and the system contribution, we discovered additional observations that might offer insights into the broader scope of research at the intersection of visualization and design optimization.

7.1 The Diversity of User Preferences

The diversity of color preferences among participants highlights the individualized nature of aesthetic value. While we can quantify many aspects of perceptually effective colormap design, taking such a theory-driven approach toward visualization aesthetics may have its limits. Instead, Cieran takes an approach that can learn individualized models of aesthetic utility given the analyst's understanding of the intended audience and data context.

Some aspects of preference may be quantifiable *a priori* with some probability. For example, people's backgrounds may impact their preferences. Our participants were researchers across the physical and social sciences. One expert participant noted that multi-hue colormaps may be preferred by those working in scientific visualization, while single-hue colormaps may be preferred by domain experts working with abstractions of data. Cultural backgrounds or gender identities may also influence preferences [54]. We may be able to leverage these factors to create foundation preference models that bootstrap optimization processes. However, they should be applied carefully to avoid perpetuating potential biases, limiting creativity, or transferring designs that call attention to context-specific data semantics to a different setting.

7.2 Exploring Then Exploiting Preferences

Some study participants specified their prior preferences for colormaps before using Cieran. While a few indicated a preference for "simple" colormaps that did not vary much in hue, others noted a prior preference for multi-hue colormaps. Still, others confessed that they lacked the experience to even have an inclination towards one colormap over the other.

Yet even those without preferences rapidly identified their own style for colormaps during our study. P1 and P7 who lacked experience with colormaps at all acknowledged that it was "very clear how color can work and how it can't" (P1) by the end of the study. P7 described Cieran as being "insightful and educational for [them], and understanding my taste and preferences for general communication," especially when making their pairwise choices. While P6 did not indicate known prior preferences and had some experience with conventional multi-hue colormaps to visualize astronomical data, they would go on to only select colormaps with low contrast values that they described as "monochromatic" or "subdued" and indicated that "simplicity" was visually appealing to them.

These observations indicate that in addition to reducing the work required to find effective designs, active preference learning tools like Cieran can help people develop a 'taste' for what they are seeking in the first place. This design exploration phase may also be interpreted through the lens of traditional visual data exploration. Different colormaps highlight different features in data [13, 57, 70]. Using active preference learning tools like Cieran, people can intelligently explore many visualization alternatives, each optimized for highlighting different data patterns. Although analysts may eventually end up finding a version that looks 'just right,' we speculate that much as datasets are often represented using multiple chart techniques, analysts may benefit from using multiple colormaps.

7.3 Diverse Inclinations Toward Agency

Effective human-AI collaboration must balance automation with agency to effectively leverage the strengths of both people and algorithms [25]. However, this balance may vary across users.

Participants wanted different levels of control over the output colormaps. Some participants indicated a lack of confidence in judging the quality of colormapped visualizations. They also questioned their ability to make internally consistent choices (P3, P6, P9) and worried about their own biases (E2). These participants expressed a desire for greater automation such as the system making choices for them based on patterns in the data. However, they still saw Cieran as an improvement on tools that only provided a fixed set of predefined palettes, and noted its efficacy in automating the process of selecting colormaps instead of manually sifting through online galleries of options (P4, P5, P6)

On the other hand, some participants wanted more creative control. For example, P7 asked to see all ranked options to make a final selection, an available feature in the tool that was not part of the user study (Figure 2). Some participants desired an interface to tweak the component colors (P1, E1, P8) or choose the interpolation technique (E1, P8). The conflict between the desire for more automation (e.g., to overcome perceived time or expertise limitations) and the desire for more control (e.g., to enhance and refine Cieran's outputs) across participants offers insight into how different paradigms for automating visualization design can serve the needs of divergent target user populations.

8 FUTURE WORK

Our system is limited by the number of expert-designed colormaps in the corpus. Furthermore, displays have a limited color gamut. When colormaps are rotated and/or translated to fit a seed color, Cieran discards some colormap trajectories because they run outside of the gamut and produce colors that many monitors cannot display. This gamut constraint motivated us to use the Tableau10 color palette when limiting the selection of seed colors in the user study since colormaps aligned to these colors were well within the gamut boundaries. Although Tableau10 was designed by a visualization expert, we plan to extend the colormap corpus to accommodate as many seed colors as possible. We also plan to compile a recommended list of 'good' seed colors.

Future iterations of Cieran could consider alternative dimensions of design. At its extreme, the algorithm could leverage techniques that learn feature functions [31] rather than relying on a prespecified feature set. However, deep learning approaches will be much slower and require an order of magnitude more comparison data to successfully model user preferences. P2, P4, and P5 also raised concerns about possible tensions between user preferences and accessibility. While our approach offers a paradigm that can account for an individual's color vision dynamically, incorporating colorblindness simulations and/or measures to Cieran would allow a more direct path toward addressing accessibility.

Lastly, while Cieran can rank and create variants of both single-hue and multi-hue colormaps effectively, users should be aware of performance trade-offs between these colormap categories for specific data domain tasks, such as spatial data analysis [13] and model checking [58]. Like many visualization construction tools, Cieran assumes that its users have an understanding of their target context, and its algorithms are not guardrails to prevent the construction of potentially inadequate visualizations: it simply acts on user inputs to help users quickly create visualizations that meet universal perceptual guidelines. Future iterations of Cieran could provide additional scenario-driven support, such as automatically adjusting hue variance based on a set of target tasks or privileging hues that align with domain conventions in addition to seed colors.

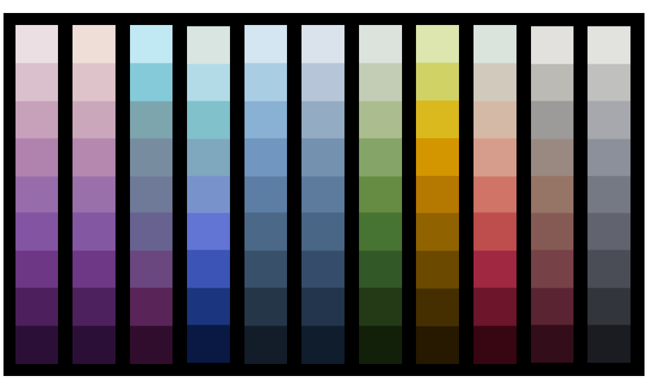

Figure 9: A set of novel colormaps created by Cieran. Each colormap above was found by Cieran using Algorithm 2 according to a study participant's preferences, and voted by the participant to be the best across the four options shown to them (Section 6.1.4).

9 CONCLUSION

To help domain experts customize colormaps optimized to their target context, our system Cieran leverages active preference learning to model the aesthetic utility of expert-designed colormap curves to rank hundreds of design options available to users (§5.3). Cieran is also able to rapidly find novel colormaps (§5.4) that are highly useful (§6.2.1). While grounded in expert examples, our system also processes colormaps to meet known best practices in colormap design. Specifically, our system leverages a preference-based reward learning algorithm, reinforcement learning, and a Jupyter Widgets interface for Python users to optimize their charts *in situ*, regardless of which plotting library was used. Our evaluation demonstrated the efficacy of this paradigm, which entailed asking only a simple question like "Which chart do you like better?" for optimizing the look-and-feel of a chart across the complex space of colormap design. We hope that this work can inspire the development of future design automation tools for data visualization.

ACKNOWLEDGMENTS

This work was supported by NSF awards #1764089, #1764092 & #2046725.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*. 1.

[2] Jarryullah Ahmad, Elaine Huynh, and Fanny Chevalier. 2021. When red means good, bad, or Canada: exploring people's reasoning for choosing color palettes. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 56–60.

[3] Riad Akrour, Marc Schoenauer, and Michèle Sebag. 2012. April: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*. Springer, 116–131.

[4] Lyn Bartram, Abhisekh Patra, and Maureen Stone. 2017. Affective color in visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1364–1374.

[5] Lawrence D Bergman, Bernice E Rogowitz, and Lloyd A Treinish. 1995. A rule-based tool for assisting colormap selection. In *Proceedings Visualization'95*. IEEE, 118–125.

[6] Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. 2007. Active Preference Learning with Discrete Choice Data. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. Curran Associates Inc., Red Hook, NY, USA, 409–416.

[7] Molly Brown. 2021. *Tableau Research leader and resident color expert Maureen Stone talks R&D at Tableau.* https://www.tableau.com/blog/how-maureen-stone-makes-data-more-accessible-tableau-research Accessed: 02-06-2023.

[8] Roxana Bujack, Terece L. Turton, Francesca Samsel, Colin Ware, David H. Rogers, and James Ahrens. 2018. The Good, the Bad, and the Ugly: A Theoretical Framework for the Assessment of Continuous Colormaps. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (jan 2018), 923–933. https://doi.org/10.1109/TVCG.2017.2743978

[9] Erdem Bıyık, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. 2019. Asking Easy Questions: A User-Friendly Approach to Active Reward Learning. In *Proceedings of the 3rd Conference on Robot Learning*. arXiv:1910.04365 http://arxiv.org/abs/1910.04365

[10] Erdem Bıyık, Aditi Talati, and Dorsa Sadigh. 2022. APReL: A Library for Active Preference-based Reward Learning Algorithms. In *ACM/IEEE International Conference on Human-Robot Interaction*, Vol. 2022-March. 613–617. https://doi.org/10.1109/HRI53351.2022.9889650 arXiv:2108.07259

[11] Qing Chen, Fuling Sun, Xinyue Xu, Zui Chen, Jiazhe Wang, and Nan Cao. 2021. Vizlinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2021), 206–216.

[12] Weiwei Cheng, Johannes Fürnkranz, Eyke Hüllermeier, and Sang-Hyeun Park. 2011. Preference-based policy iteration: Leveraging preference learning for reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*. Springer, 312–327.

[13] Aritra Dasgupta, Jorge Poco, Bernice Rogowitz, Kyungsik Han, Enrico Bertini, and Claudio T Silva. 2018. The effect of color scales on climate scientists' objective and subjective performance in spatial data analysis tasks. *IEEE Transactions on Visualization and Computer Graphics* 26, 3 (2018), 1577–1591.

[14] Carl de Boor. 1972. On calculating with B-splines. *Journal of Approximation Theory* 6, 1 (jul 1972), 50–62. https://doi.org/10.1016/0021-9045(72)90080-9

[15] Eyal Even-Dar, Yishay Mansour, and Yishay Mansourt. 2001. Convergence of Optimistic and Incremental Q-Learning. In *Advances in Neural Information Processing Systems*, T Dietterich, S Becker, and Z Ghahramani (Eds.), Vol. 14. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2001/file/6f2688a5fce7d48c8d19762b88c32c3b-Paper.pdf

[16] Roland Geraerts and Mark H. Overmars. 2004. A Comparative Study of Probabilistic Roadmap Planners. In *Algorithmic Foundations of Robotics V*. Springer, 43–57. https://doi.org/10.1007/978-3-540-45058-0_4

[17] Connor C. Gramazio, David H. Laidlaw, and Karen B. Schloss. 2017. Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (jan 2017), 521–530. https://doi.org/10.1109/TVCG.2016.2598918

[18] D. A. Green. 2011. A colour scheme for the display of astronomical intensity images. *Bulletin of the Astronomical Society of India* 39, 2 (aug 2011), 289–295. arXiv:1108.5083 http://arxiv.org/abs/1108.5083

[19] Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. *Exploring network structure, dynamics, and function using NetworkX*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[20] J. H. Halton. 1960. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* 2, 1 (dec 1960), 84–90. https://doi.org/10.1007/BF01386213

[21] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[22] Mark Harrower and Cynthia A. Brewer. 2003. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal* 40, 1 (2003), 27–37. https://doi.org/10.1179/000870403235002042 arXiv:https://www.tandfonline.com/doi/pdf/10.1179/000870403235002042

[23] Tingying He, Petra Isenberg, Raimund Dachselt, and Tobias Isenberg. 2022. BeauVis: A Validated Scale for Measuring the Aesthetic Pleasure of Visual Representations. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2022), 363–373.

[24] Christopher Healey, Sarat Kocherlakota, Vivek Rao, Reshma Mehta, and Robert St Amant. 2008. Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE Transactions on Visualization and Computer Graphics* 14, 2 (2008), 396–411.

[25] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850. https://doi.org/10.1073/pnas.1807184115 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1807184115

[26] Aspen K Hopkins, Michael Correll, and Arvind Satyanarayan. 2020. Visualint: Sketchy in situ annotations of chart construction errors. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 219–228.

[27] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[28] Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. 1994. On the Convergence of Stochastic Iterative Dynamic Programming Algorithms. *Neural Computation* 6, 6 (11 1994), 1185–1201. https://doi.org/10.1162/neco.1994.6.6.1185 arXiv:https://direct.mit.edu/neco/article-pdf/6/6/1185/812885/neco.1994.6.6.1185.pdf

[29] Jupyter Widgets Team. 2023. ipywidgets: Interactive Widgets for the Jupyter Notebook. https://github.com/jupyter-widgets/ipywidgets GitHub repository.

[30] Daniel Kahneman and Amos Tversky. 1979. Prospect Theory: An Analysis of Decision under Risk. *Econometrica* 47, 2 (mar 1979), 263. https://doi.org/10.2307/1914185

[31] Sydney M. Katz, Anne Claire Le Bihan, and Mykel J. Kochenderfer. 2019. Learning an Urban Air Mobility Encounter Model from Expert Preferences. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, Vol. 2019-Septe. https://doi.org/10.1109/DASC43569.2019.9081648 arXiv:1907.05575

[32] W Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*. 9–16.

[33] Yuki Koyama and Masataka Goto. 2022. BO as Assistant: Using Bayesian Optimization for Asynchronously Generating Design Suggestions. *UIST 2022 - Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (2022). https://doi.org/10.1145/3526113.3545664

[34] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics* 36, 4 (aug 2017), 1–11. https://doi.org/10.1145/3072959.3073598

[35] K. S. Krishnan. 1977. Incorporating Thresholds of Indifference in Probabilistic Choice Models. *Management Science* 23, 11 (1977), 1224–1233. http://www.jstor.org/stable/2630661

[36] Andrea Lau and Andrew Vande Moere. 2007. Towards a model of information aesthetics in information visualization. In *2007 11th International Conference Information Visualization (IV'07)*. IEEE, 87–92.

[37] Shuqi Liu, Mingtian Tao, Yifei Huang, Changbo Wang, and Chenhui Li. 2022. Image-Driven Harmonious Color Palette Generation for Diverse Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–16. https://doi.org/10.1109/TVCG.2022.3226218

[38] Yang Liu and Jeffrey Heer. 2018. Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174172

[39] S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (mar 1982), 129–137. https://doi.org/10.1109/TIT.1982.1056489

[40] R.Duncan Luce. 1977. The Choice Axiom After Twenty Years. *Journal of Mathematical Psychology* 15, 3 (jun 1977), 215–233. https://doi.org/10.1016/0022-2496(77)90032-3

[41] M. R. Luo, G. Cui, and B. Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research and Application* 26, 5 (oct 2001), 340–350. https://doi.org/10.1002/col.1049

[42] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1137–1144.

[43] Joel Max. 1960. Quantizing for minimum distortion. *IEEE Transactions on Information Theory* 6, 1 (mar 1960), 7–12. https://doi.org/10.1109/TIT.1960.1057548

[44] Andrew McNutt and Gordon Kindlmann. 2018. Linting for visualization: Towards a practical automated visualization guidance system. In *VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization*. 1–14.

[45] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21, 6 (jun 1953), 1087–1092. https://doi.org/10.1063/1.1699114

[46] Sebastian Mittelstädt, Dominik Jäckle, Florian Stoffel, and Daniel A. Keim. 2015. ColorCAT: Guided Design of Colormaps for Combined Analysis Tasks. *Eurographics Conference on Visualization, EuroVis 2015 - Short Papers* (2015), 115–119. https://doi.org/10.2312/eurovisshort.20151135

[47] Wojciech Mokrzycki and Maciej Tatol. 2011. Color difference Delta E - A survey. *Machine Graphics and Vision* 20 (04 2011), 383–411.

[48] Kenneth Moreland. 2009. Diverging Color Maps for Scientific Visualization. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II* (Las Vegas, Nevada) *(ISVC '09)*. Springer-Verlag, Berlin, Heidelberg, 92–103. https://doi.org/10.1007/978-3-642-10520-3_9

[49] Kenneth D Moreland. 2015. *Why We Use Bad Color Maps and What You Can Do About It*. Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[50] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2018), 438–448.

[51] Isaac Muse. 2023. coloraide: A Python library for color. https://github.com/facelessuser/coloraide GitHub repository.

[52] Pascal Nardini, Min Chen, Michael Böttinger, Gerik Scheuermann, and Roxana Bujack. 2021. Automatic Improvement of Continuous Colormaps in Euclidean Colorspaces. *Computer Graphics Forum* 40, 3 (2021), 361–373. https://doi.org/10.1111/cgf.14313

[53] Pascal Nardini, Min Chen, Francesca Samsel, Roxana Bujack, Michael Bottinger, and Gerik Scheuermann. 2019. The Making of Continuous Colormaps. *IEEE Transactions on Visualization and Computer Graphics* 14, 8 (2019), 1–1. https://doi.org/10.1109/tvcg.2019.2961674

[54] Stephen E. Palmer, Karen B. Schloss, and Jonathan Sammartino. 2013. Visual Aesthetics and Human Preference. *Annual Review of Psychology* 64, 1 (jan 2013), 77–107. https://doi.org/10.1146/annurev-psych-120710-100504

[55] Ghulam Jilani Quadri, Jennifer Adorno Nieves, Brenton M. Wiernik, and Paul Rosen. 2023. Automatic Scatterplot Design Optimization for Clustering Identification. *IEEE Transactions on Visualization and Computer Graphics* 29, 10 (2023), 4312–4327. https://doi.org/10.1109/TVCG.2022.3189883

[56] Ghulam Jilani Quadri and Paul Rosen. 2021. Modeling the Influence of Visual Density on Cluster Perception in Scatterplots Using Topology. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1829–1839. https://doi.org/10.1109/TVCG.2020.3030365

[57] Khairi Reda. 2022. Rainbow Colormaps: What are they good and bad for? *IEEE Transactions on Visualization and Computer Graphics* (2022).

[58] Khairi Reda and Danielle Albers Szafir. 2021. Rainbows Revisited: Modeling Effective Colormap Design for Graphical Inference. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (feb 2021), 1032–1042. https://doi.org/10.1109/TVCG.2020.3030439

[59] Dorsa Sadigh, Anca D. Dragan, Shankar Sastry, and Sanjit A. Seshia. 2017. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, Vol. 13. MIT Press Journals. https://doi.org/10.15607/rss.2017.xiii.053

[60] Francesca Samsel, Sebastian Klaassen, and David H. Rogers. 2018. ColorMoves: Real-time Interactive Colormap Construction for Scientific Visualization. *IEEE Computer Graphics and Applications* 38, 1 (jan 2018), 20–29. https://doi.org/10.1109/MCG.2018.011461525

[61] Karen B Schloss, Zachary Leggon, and Laurent Lessard. 2020. Semantic discriminability for visual communication. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1022–1031.

[62] Stephen Smart, Keke Wu, and Danielle Albers Szafir. 2020. Color Crafting: Automating the Construction of Designer Quality Color Ramps. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (aug 2020), 1215–1225. https://doi.org/10.1109/TVCG.2019.2934284 arXiv:1908.00629

[63] Nathaniel J. Smith and Stefan van der Walt. 2019. viscm. https://github.com/matplotlib/viscm.

[64] Sicheng Song, Chenhui Li, Yujing Sun, and Changbo Wang. 2023. VividGraph: Learning to Extract and Redesign Network Graphs From Visualization Images. *IEEE Transactions on Visualization and Computer Graphics* 29, 7 (2023), 3169–3181. https://doi.org/10.1109/TVCG.2022.3153514

[65] Maureen Stone. 2016. How we designed the new color palettes in Tableau 10. https://www.tableau.com/blog/colors-upgrade-tableau-10-56782?hootPostID=4574bce614837b6f773ae492b0c516f7. Accessed: 2023-04-03.

[66] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

[67] Danielle Albers Szafir. 2017. Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2017), 392–401.

[68] Robert Endre Tarjan. 1978. Complexity of Combinatorial Algorithms. *SIAM Rev.* 20, 3 (jul 1978), 457–491. https://doi.org/10.1137/1020067

[69] Andrea L Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6-7 (2008), 716–737.

[70] Christian Tominski, Georg Fuchs, and Heidrun Schumann. 2008. Task-driven color coding. In *2008 12th International Conference Information Visualisation*. IEEE, 373–380.

[71] Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. 2018. Altair: interactive statistical visualizations for Python. *Journal of Open Source Software* 3, 32 (2018), 1057.

[72] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[73] Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software* 6, 60 (2021), 3021. https://doi.org/10.21105/joss.03021

[74] Christopher John Cornish Hellaby Watkins. 1989. *Learning from delayed rewards*. PhD Thesis. Kings College, University of Cambridge,. http://www.cs.rhul.ac.uk/$\sim$chrisw/new_thesis.pdf

[75] Christopher John Cornish Hellaby Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* 8, 3-4 (may 1992), 279–292. https://doi.org/10.1007/bf00992698

[76] Lin-Ping Yuan, Wei Zeng, Siwei Fu, Zhiliang Zeng, Haotian Li, Chi-Wing Fu, and Huamin Qu. 2022. Deep Colormap Extraction From Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2022), 4048–4060. https://doi.org/10.1109/TVCG.2021.3070876

[77] Lin-Ping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, and Huamin Qu. 2022. InfoColorizer: Interactive Recommendation of Color Palettes for Infographics. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2022), 4252–4266. https://doi.org/10.1109/TVCG.2021.3085327

[78] Qian Zheng, Min Lu, Sicong Wu, Ruizhen Hu, Joel Lanir, and Hui Huang. 2022. Image-guided color mapping for categorical data visualization. *Computational Visual Media* 8, 4 (2022), 613–629.

[79] Clementine Zimnicki, Chin Tseng, Danielle Albers Szafir, and Karen B Schloss. 2023. Effects of data distribution and granularity on color semantics for colormap data visualizations. In *2023 IEEE Visualization and Visual Analytics - Short Papers*. https://doi.org/10.1109/VIS54172.2023.00011

A UNIFORM PARAMETRIZATION OF CURVES BY ARC LENGTH

A curve can be defined as a parametric function $C(t)$, where t is parameter point in the continuous interval $[0, 1]$. Our goal is to find a discrete and increasing parameter set $\{t_0^*, \cdots, t_N^*\}$ such that the distance between any two consecutive coordinates $D(C(t_i^*), C(t_{i+1}^*))$ is the same.

The following algorithm was also used to design perceptually uniform `matplotlib` colormaps (magma, inferno, plasma, and viridis) [63]. However, these colormaps utilize simple Euclidean distances between sampled colors in the CIELAB colorspace. In contrast, Cieran uses the ΔE_{2000} perceptual distance metric [41] for D.

Given an initial parameter set $\{t_0, \cdots, t_N\}$ evenly sampled across $[0, 1]$, we first compute the cumulative rectified arc lengths at each t_i as follows:

$$s(t_i) = \sum_{k=1}^{i} D(C(t_i), C(t_{i+1})), \qquad \text{for } i = 1, 2, \ldots, N$$

with the initial condition that:

$$s(t_0) = 0$$

Dividing each $s(t_i)$ by the total arc length $s(t_N)$ gives us the relative cumulative arc length at each parameter point:

$$\hat{s}(t_i) = \frac{s(t_i)}{s(t_N)}$$

and allows $\hat{s}(t_i)$ and t_i to share the same domain $[0, 1]$. We then approximate $\hat{s}^{-1}$ that satisfies $\hat{s}^{-1}(\hat{s}(t_i)) = t_i$ by inversely mapping each relative cumulative arc length to a parameter point and performing interpolation.

Finally, we obtain new parameter points by applying the interpolated $\hat{s}^{-1}$ to evenly spaced values within $[0, 1]$. Although the resulting outputs $\{t_0^*, \cdots, t_N^*\}$ will no longer be evenly spaced within $[0, 1]$, their corresponding coordinates $\{C(t_0^*), \ldots, C(t_N^*)\}$ will be evenly spaced in terms of rectified arc lengths $D(C(t_i), C(t_{i+1})) = \Delta E_{2000}(C(t_i), C(t_{i+1}))$ (Figure 10).

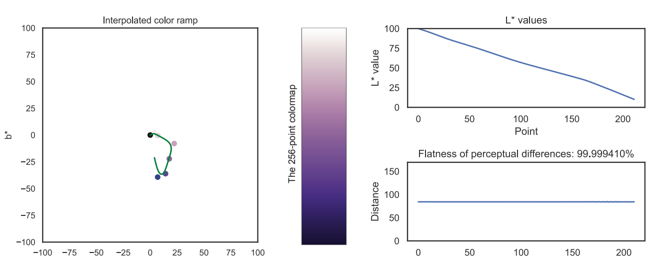

Figure 10: Profile of a colormap created by Cieran. Each colormap output by Cieran will be smooth, monotonically varying in lightness and perceptually uniform. The bottom right subplot represents the perceptual distance between each adjacent pair of colors in the 256-point continuous colormap. The "flatness" metric is 1.0 minus the std. dev. of distances divided by total arc length.

B ADDITIONAL QUALITATIVE FINDINGS

We further explored the robustness of Cieran toward different user-specified seed colors in our Evaluation. Figure 11 represents the distribution of 1st and 2nd place votes in the final ranking task in the study. Each bar corresponds to a colormap category (new, top-ranked, median-ranked, last-ranked). We can verify that the majority of 1st and 2nd place votes went to the new and top-ranked colormaps for every seed color.

However, our study comprised 36 total design sessions, and not all colors were selected with equal frequency. Bar groups have been sorted based on the frequency with which that seed color was selected. Although participants that initialized Cieran with the bottom two colors (gray and yellow) never gave 1st and 2nd place votes to median- and last-ranked colormaps, the sample sizes for these two colors are each one. Beginning from the top color (blue), each seed color was selected with the following frequency:

- blue: 7
- teal: 6
- green: 6
- purple: 5
- orange: 4
- red: 3
- brown: 3
- gray: 1
- yellow: 1

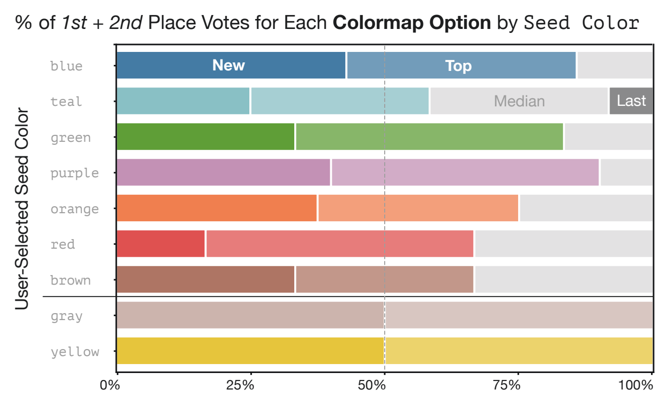

Figure 11: Summary of participant rank responses from the user study as a function of their initial seed color choice. Cieran performs well across a variety of seed colors.