

Supplementary Material for C2Views: Knowledge-based Colormap Design for Multiple-View Consistency

Appendix A: Implementation Details

Pseudocode

Algorithm 1: Pareto-based Genetic Algorithm for Colormap Optimization

Input: Population size N , Max generations G , View specifications V , Initial seed palette S

Output: A set of Pareto-optimal colormap solutions \mathcal{P}_{final}

```

1  $P_0 \leftarrow \text{INITIALIZEPOPULATION}(N, V, S)$ 
2  $t \leftarrow 0$ 
3 while  $t < G$  do
4    $C_{set} \leftarrow \emptyset$ 
5   foreach individual  $m$  in  $P_t$  do
6     Compute fitness score  $(C_{SV}(m), C_{MV}(m))$ 
7      $C_{set} \leftarrow C_{set} \cup \{(m, [C_{SV}(m), C_{MV}(m)])\}$ 
8    $\mathcal{P}_t \leftarrow \text{FINDPARETOFRONT}(C_{set})$ 
9    $P_{next} \leftarrow \{m \mid (m, \cdot) \in \mathcal{P}_t\}$  // Carry over elite solutions
10  while  $|P_{next}| < N$  do
11     $(A, B) \leftarrow \text{SELECTPARENTS}(\mathcal{P}_t)$  // Randomly sample
12     $(O_1, O_2) \leftarrow \text{CROSSOVER}(A, B)$ 
13     $(O_1, O_2) \leftarrow \text{MUTATE}(O_1, O_2)$ 
14     $P_{next} \leftarrow P_{next} \cup \{O_1, O_2\}$ 
15   $P_{t+1} \leftarrow P_{next}$ 
16   $t \leftarrow t + 1$ 
17  $\mathcal{P}_{final} \leftarrow \text{POSTPROCESS}(\mathcal{P}_t)$ 
18 return  $\mathcal{P}_{final}$ 

```

Metrics and Parameters

To ensure the reproducibility of our optimization framework, this section provides the implementation details regarding our metric handling, key parameters, and threshold settings.

Normalization of Metrics. Our cost function consists of four components. To balance their magnitudes, we employ Min-Max Scaling to normalize each cost term to the range of $[0, 1]$. The normalization is calculated as follows:

$$\text{norm_cost} = \frac{\text{cost} - \text{min_cost}}{\text{max_cost} - \text{min_cost}} \quad (1)$$

Here, cost is the raw value, while min_cost and max_cost are the historical minimum and maximum values recorded across multiple runs. These historical extrema are stored in a `params.json` file and are independently maintained and updated for each test case. This

dynamic updating mechanism allows the normalization to adapt to the cost distributions of different optimization tasks.

Weight Settings. The relative importance of the four cost terms can be adjusted by the user. By default, all weights are set to 1, corresponding to single-view color discriminability, color continuity, multi-view color discriminability, and hue uniformity, respectively. This initial setting treats each design criterion with equal importance.

Cost Function Thresholds and Penalty Factor. In the calculation of raw costs, we define specific thresholds to distinguish between acceptable and suboptimal color differences. When a calculated value fails to meet a threshold, we apply a smooth penalty using a Sigmoid function.

- **Penalty Factor:** A global `PENALTY_FACTOR` is set to 0.2 to adjust the steepness of the Sigmoid function.
- **Minimum Thresholds:** The predefined thresholds for the cost functions are as follows:
 - `color_difference_sv`: 30
 - `color_continuity`: 30
 - `color_difference_mv`: 20
 - `hue_uniformity`: 20
- **Fitness Evaluation Thresholds:** During the genetic algorithm's fitness evaluation, a secondary check is applied to the *normalized* cost values. The `max_allowed_values` for the four normalized cost terms are set to $[0.2, 0.2, 0.2, 0.2]$. If a solution's cost exceeds this value, an additional penalty is applied to guide the search toward solutions that perform well across all metrics.

Genetic Algorithm Parameters

Our genetic algorithm is configured with the following parameters. These are based on empirical observations.

- **Population Size:** `pop_size` = 50.
- **Number of Generations:** `generations` = 100.
- **Elite Pool Size:** `n_best` = 10. This parameter specifies the number of top-performing individuals selected from the Pareto front to serve as the primary pool for creating the next generation.
- **Selection Mechanism:** We employ an elitist selection strategy. In each generation, all non-dominated solutions are identified to form the Pareto front. From this front, we select up to `n_best`

individuals to form the elite pool. Parents for reproduction are then randomly sampled from this elite pool.

- **Crossover Rate:** 0.5. For any two parents, their color schemes for views at hierarchy level 1 are swapped with a 50% probability.
- **Mutation Step:** Mutation is performed by applying a small random perturbation to a color's HSV values. The magnitude of this perturbation is controlled by the step parameter, which defaults to 0.05 and is user-adjustable.

Appendix B: Quantitative Evaluation Metrics

This appendix provides the mathematical definitions for the three quantitative metrics used in our evaluation (Sect. 5.2). For all metrics, a higher score indicates better performance.

Worst-Case Discriminability (WCD) WCD assesses single-view clarity by measuring the minimum perceptual distance between any two colors within a single view's colormap. It quantifies the "weakest link" in color differentiability, where a higher score means even the most similar colors are easy to distinguish.

$$\text{WCD}(V) = \min_{c_i, c_j \in C(V), i \neq j} \Delta E_{00}(c_i, c_j) \quad (2)$$

Here, $C(V)$ is the set of colors in the colormap for view V . For an entire multi-view visualization, the overall WCD is the minimum WCD score across all views.

Parallel Relationship Score (PRS) PRS evaluates multi-view consistency for non-hierarchical (i.e., partial-redundancy) relationships. It measures the minimum perceptual distance between any two colors from different views that represent different data entities. A high PRS ensures that unrelated items across the visualization are not perceptually confused with one another.

$$\text{PRS}(\mathcal{V}) = \min_{\substack{c_a \in C(V_i), c_b \in C(V_j) \\ V_i, V_j \in \mathcal{V}, i \neq j \\ \text{key}(c_a) \neq \text{key}(c_b)}} \Delta E_{00}(c_a, c_b) \quad (3)$$

Here, \mathcal{V} is the set of all views in the visualization, and $\text{key}(c)$ refers to the underlying data entity that the color c encodes.

Hierarchical Quality Score (HQS) HQS assesses the quality of hierarchical colormaps by balancing two competing factors: the internal discriminability of the child colors and their hue similarity to the parent color. It is defined as the ratio of the child colors' worst-case discriminability (Child-WCD) to their average hue deviation from the parent (HHD).

$$\text{HQS} = \frac{\text{Child-WCD}}{1 + \text{HHD}} \quad (4)$$

The two components are defined as:

- **Child-WCD:** The minimum perceptual distance between any pair of colors in the child set, calculated identically to WCD.

$$\text{Child-WCD} = \min_{c_i, c_j \in C_{\text{child}}, i \neq j} \Delta E_{00}(c_i, c_j)$$

- **HHD (Hierarchical Hue Deviation):** The average absolute difference in hue between the parent color c_p and each child color

c_k , accounting for the circular nature of the hue space.

$$\text{HHD} = \frac{1}{|C_{\text{child}}|} \sum_{c_k \in C_{\text{child}}} \min(|H(c_p) - H(c_k)|, 360^\circ - |H(c_p) - H(c_k)|)$$

Here, c_p is the parent color, C_{child} is the set of child colors, and $H(c)$ is the hue of color c in degrees.

Appendix C: Extra Cases

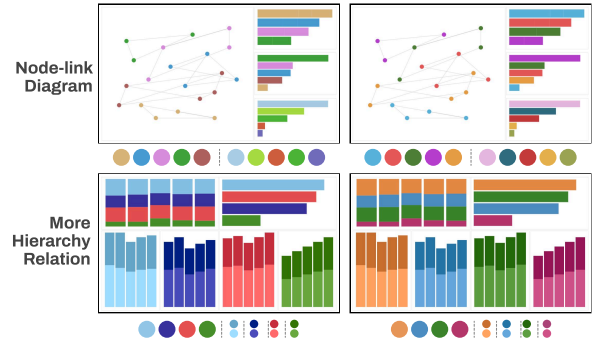


Figure 1: Extra cases including a node-link diagram and a 4 categorical hierarchical case.

Our framework is based on modeling abstract data relationships inferred from Vega-Lite specifications. This allows our approach to generalize to more supported chart types. For instance, the top case in Figure above demonstrates a successful application to a node-link diagram, maintaining color consistency with a corresponding bar chart. Additionally, we present a solution for a complex hierarchical scenario with four distinct parent categories. This case is inspired by the real-world example of poor hierarchical coloring discussed in our Introduction (Figure 1b).