Supplemental Material: Dynamic Color Assignment for Hierarchical Data

1 Datasets

Tab. S1 presents the detailed information of 12 datasets used in our qualitative evaluation, including the number of classes and the number of samples. The first six datasets are flat datasets with a moderate range of classes, which are used to evaluate flat color assignment methods. The remaining six datasets have a larger number of classes, which are used to evaluate hierarchical color assignment methods.

Table S1: The information of the datasets.

Dataset	# Classes	# Samples
MNIST [11]	10	70,000
Animals [4]	10	26,179
Indian Food [13]	13	4,770
Clothing [21]	14	37,497
Texture [1]	15	5,500
Isolet [7]	26	7,797
Food101 [2]	101	101,000
Flowers102 [14]	102	8,189
Stanford Cars [10]	196	16,185
Caltech256 [9]	257	30,607
Nabirds [18]	555	48,000
ImageNet1k [5]	1,000	1,281,167

2 Running Time

We evaluated the running time of our method on a Linux server with an Intel i9-13900K CPU (3.0 GHz). The running time consists of two parts: 1) time for computing nearest neighbors for the spatial distribution term and 2) time for optimizing color assignment results using simulated annealing. The first part depends on the dataset size. As shown in Table S2, it takes less than 0.2 seconds to compute the nearest neighbors for 100,000 samples in a scatterplot. The second part is independent of the dataset size but depends on the number of classes. As shown in Table S3, our algorithm can assign colors to 20 classes within 0.4 seconds and to 30 classes within 1 second. Previous research has shown that a color palette with 26 colors already exhibits poor discriminability [6]. Therefore, our method well supports real-time interaction for users navigating through hierarchical visualizations. Table S3 further details the processing times for different stages. Most of the time is spent on the first stage to enhance discriminability, with subsequent optimization stages achieving rapid convergence.

3 Additional Color Assignment Results

In this section, we offer four additional examples that compare hierarchical color assignment results, each corresponding to one of the four visualization types.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Table S2: Average running time (in seconds) of computing nearest neighbors in scatterplots with different numbers of samples.

#Sample	1,000	10,000	100,000
Time	0.002	0.014	0.171

Table S3: Average running time (in seconds) of optimizing color assignment results in each stage.

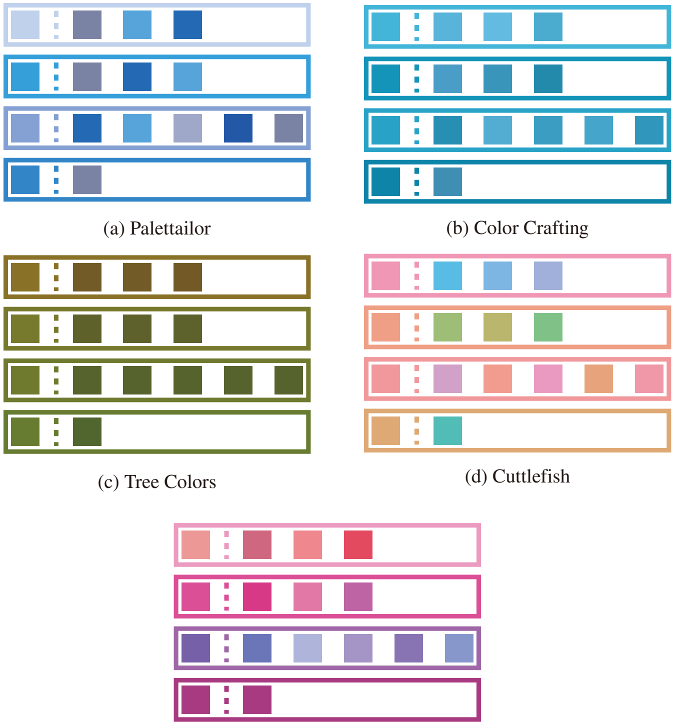
Stages	10 classes	20 classes	30 classes
Total	0.063	0.352	0.968
Discriminability	0.039	0.239	0.655
Harmony	0.018	0.098	0.258
Spatial Distribution	0.006	0.015	0.055

Palette. The generation of the palette does not incorporate the spatial distribution information of the data. As users explore, the colors corresponding to child classes are generated solely based on the colors of their parent classes. Fig. S1 presents an example where users explore a lower hierarchical level. The colors on the left side of the dividing line are the colors of the parent classes, while the colors on the right side are the colors of their child classes. It can be observed that Color Crafting [16] and Tree Colors [17] produce many similar colors, resulting in poor discriminability. Meanwhile, Palettailor [12] and Cuttlefish [19] perform poorly in the alignment with hierarchy, making it difficult for users to associate parent and child classes. Our approach improves discriminability and harmony while ensuring consistency, yielding relatively satisfactory results.

Scatterplot. We adopt the approach proposed by Xiang *et al*. [20] to generate hierarchical scatterplots. It employs incremental t-SNE to dynamically generate scatterplots for each layer. We then apply the color assignment results generated by different methods. Fig. S2 presents an example where users explore a higher hierarchical level. It can be observed that almost all baseline methods do not perform sufficiently well in discriminability, with colors that are difficult to distinguish. While Palettailor performs relatively well, there are still some colors that cannot be differentiated. In contrast, our method ensures discriminability and improves color harmony.

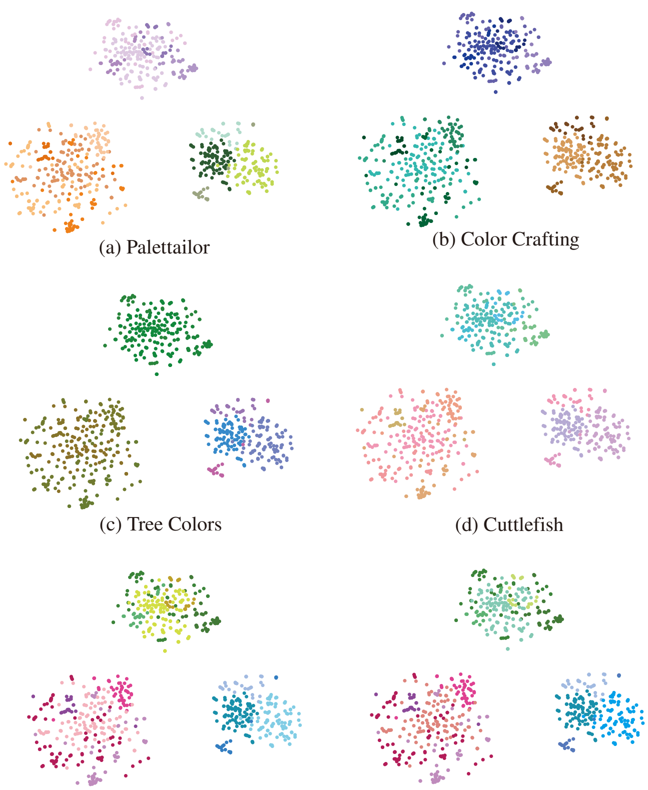
Parallel coordinates. We adopt the approach proposed by Fua *et al*. [8] to generate hierarchical parallel coordinates. Fig. S3 presents an example where users explore a higher hierarchical level. The high quantity and complexity of the lines demand high discriminability for quality color assignment. Our method is relatively well-suited to adapt to such situations.

Grid visualization. We utilized the approach proposed by Chen [3] to generate hierarchical grid visualizations. Compared to scatterplots and parallel coordinates, grid visualizations using surfaces can more easily display colors. Fig. S4 presents an example where users explore a higher hierarchical level. It can be seen that our method exhibits relatively high discriminability and harmony. In comparison, the result of Cuttlefish [19] is harmonic but contains very similar colors.



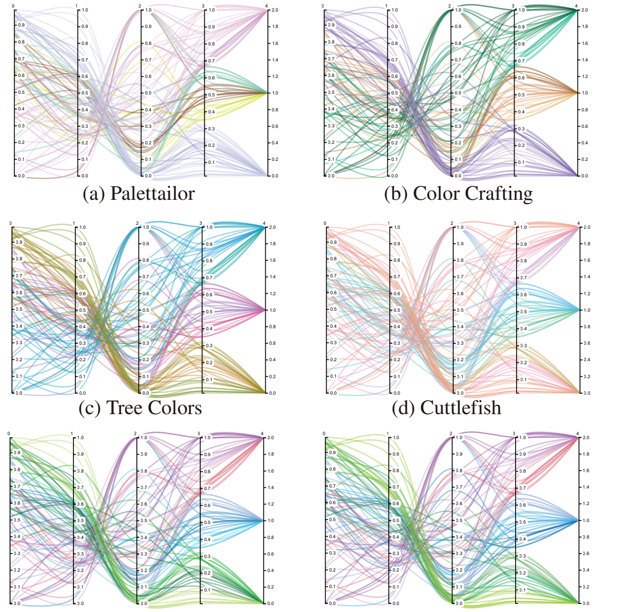
(a) Palettailor (b) Color Crafting (c) Tree Colors (d) Cuttlefish (e) Ours-D/Ours-S

Fig. S1: An example of exploring a lowerw hierarchical level in the form of the color palette. Since spatial distribution is not considered, Ours-S and Ours-D generate the same result.



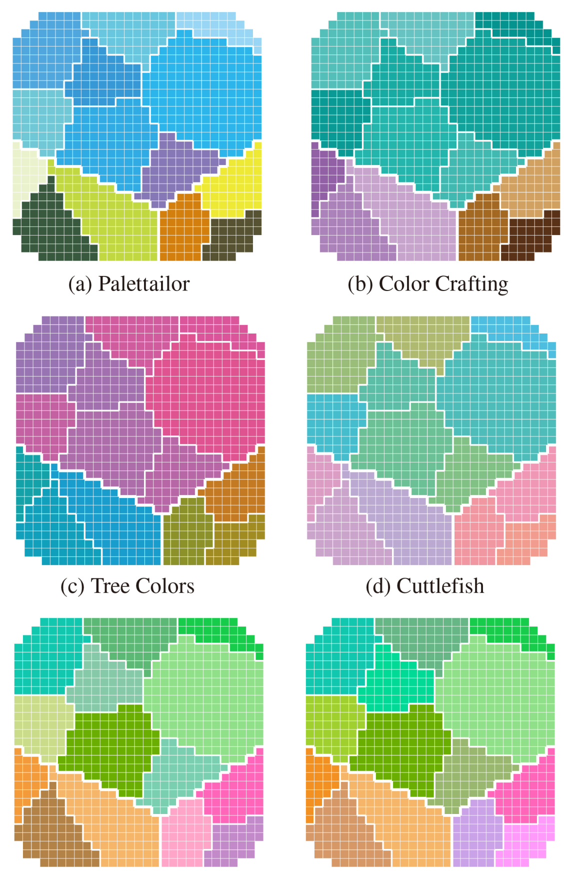
(a) Palettailor (b) Color Crafting (c) Tree Colors (d) Cuttlefish (e) Ours-D (e) Ours-S

Fig. S2: An example of exploring a higher hierarchical level in the form of scatterplot.



(a) Palettailor (b) Color Crafting (c) Tree Colors (d) Cuttlefish (e) Ours-D (e) Ours-S

Fig. S3: An example of exploring a higher hierarchical level in the form of parallel coordinates.



(a) Palettailor (b) Color Crafting (c) Tree Colors (d) Cuttlefish (e) Ours-D (e) Ours-S

Fig. S4: An example of exploring a higher hierarchical level in the form of grid visualization.

Table S4: Comparison of our method with the representative color assignment methods for flat color assignment, with the best in **bold** and the second best underlined. The values in gray indicate that the colors cannot be easily distinguished.

PD: perceptual difference, ND: name difference, ACIE76: average CIE76 color difference, MCIE76: minimal CIE76 color difference, CL: chroma-lightness, BHDI: balanced harmony-discrimination index.

Methods	Flat color assignment						
	Discriminability				Harmony		BHDI
	PD	ND	ACIE76	MCIE76	Hue	CL	
Palettailor	19.419	0.913	**96.696**	30.104	0.296	0.377	4.441
Color Crafting	2.549	0.261	25.253	4.548	**1.000**	**1.000**	2.777
Tree Colors	6.168	0.848	75.599	11.903	0.608	**1.000**	3.909
Cuttlefish	9.944	0.813	54.549	18.005	0.702	**1.000**	4.323
Ours-D	**23.194**	**0.921**	<u>79.968</u>	**30.594**	0.876	0.955	<u>5.992</u>
Ours-S	<u>23.070</u>	<u>0.920</u>	79.742	<u>30.495</u>	<u>0.893</u>	<u>0.962</u>	**6.002**

Table S5: Comparison of our method with the representative color assignment methods for hierarchical color assignment, with the best in **bold** and the second best underlined. The values in gray indicate that the colors cannot be easily distinguished.

PD: perceptual difference, ND: name difference, ACIE76: average CIE76 color difference, MCIE76: minimal CIE76 color difference, CL: chroma-lightness, BHDI: balanced harmony-discrimination index, SS: silhouette score, DR: distance ratio, RoMM: ratio of the maximum color distance between siblings to the minimum color distance between cousins.

Methods	Hierarchical color assignment									
	Discriminability				Harmony		BHDI	Alignment with hierarchy		
	PD	ND	ACIE76	MCIE76	Hue	CL		SS	DR	RoMM
Palettailor	8.202	0.508	<u>51.697</u>	16.102	0.838	0.579	3.253	0.582	0.883	6.471×10^5
Color Crafting	4.176	0.243	<u>23.015</u>	8.143	**0.997**	<u>0.998</u>	2.898	0.642	0.930	3.038
Tree Colors	0.226	0.066	6.959	0.379	0.978	**1.000**	2.133	**0.785**	**0.994**	4.370×10^3
Cuttlefish	4.615	0.382	25.248	8.497	0.606	**1.000**	2.832	0.715	0.907	0.972
Ours-D	**16.579**	**0.736**	**53.552**	**23.862**	0.984	0.810	<u>4.926</u>	<u>0.740</u>	0.945	**0.527**
Ours-S	<u>16.482</u>	<u>0.699</u>	48.635	<u>20.852</u>	<u>0.985</u>	0.927	**4.958**	<u>0.740</u>	<u>0.946</u>	<u>0.538</u>

4 Quantitative Evaluation with Additional Measures

In this section, we present the comparison results using three additional measures: average CIE76 color difference (ACIE76), minimal CIE76 color difference (MCIE76), and the ratio of the maximum color distance between siblings to the minimum color distance between cousins (RoMM). ACIE76 and MCIE76 are commonly used measures for color discriminability [15] and are not used in our optimization process. RoMM evaluates the alignment with hierarchy, where a lower RoMM indicates that the colors under the same parents have more similar colors.

Tables S4 and S5 provide a comparative analysis of the flat and hierarchical color assignment, respectively. In the flat color assignment, Ours-D performs the best in MCIE76 and ranks second to Paletailor [12] in ACIE76. However, it should be noted that Paletailor achieves its high ACIE76 score by using a much broader chroma range ($[0, 100]$ *vs.* ours $[40, 85]$). This broader range results in the inclusion of extreme colors, such as dim colors with low chroma (see colors ■ ■ in Fig S4(a)). Consequently, Paletailor records the lowest harmony scores, with 0.296 in Hue and 0.377 in CL, indicating a significant compromise in color harmony. In the hierarchical color assignment, Ours-D performs the best in all three additional measures. Cuttlefish [19] also shows better performance in RoMM by maintaining hue consistency between colors of the same parent. However, it still performs worse than our methods, which employ constraints in dynamic range selection to ensure color compactness under the same parent and sufficient separation across different parents. In addition, Cuttlefish may still produce similar colors when there are too many subclasses due to the limited use of lightness and chroma (*e.g.*, colors ■ ■ in Fig S4(d)). This limitation is also evidenced by its lower scores in discriminability (4.615 in PD and 8.497 in MCIE76).

5 Additional Results of User Study

In this section, we provide additional results from the user study, including the distribution of the ranks and frequency of the methods that are ranked first. Similar conclusions to those obtained in the paper can be drawn based on these graphical results.

Distribution of the ranks. As shown in Fig. S5, it can be seen that, overall, our method performs the best, followed by Cuttlefish [19]. This indicates the advantage of dynamic methods in handling data under hierarchy. Fig. S6 shows the performance on the discriminability task, where our method outperforms all other methods, with Palettailor [12] coming next as it focuses on optimizing distinctiveness. Fig. S7 demonstrates the performance on the harmony task, where our method approaches Cuttlefish and outperforms other methods. Fig. S8 illustrates the performance on the alignment with the hierarchy task, where our method outperforms all other methods.

Ratios of methods that are ranked highest. In addition, we analyze which methods users most often rank as their first choice across various trials. The results are summarized in Figs. S9 to S12. It is observed that users generally prefer our method and rank it as their first choice. Regarding harmony, Cuttlefish, and our method are comparable, which is consistent with the findings we draw earlier. Specifically, as illustrated in Fig. S12, our method demonstrates excellent alignment with the hierarchy even at the lower level.

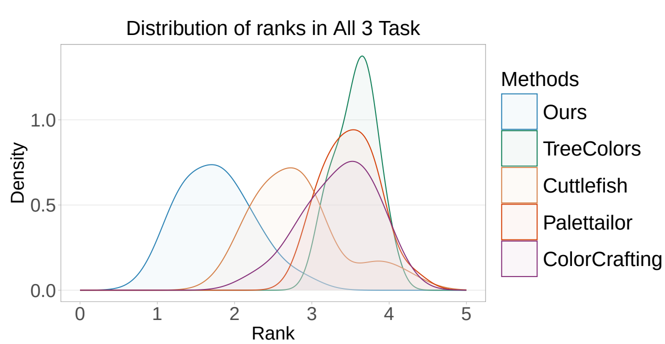


Fig. S5: The distribution of the ranks of different methods on all the three tasks

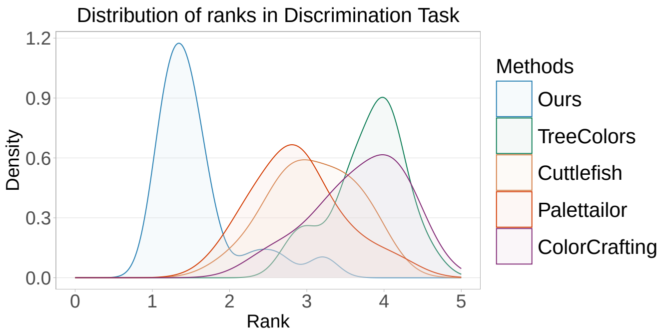


Fig. S6: The distribution of the ranks of different methods on the task of discriminability.

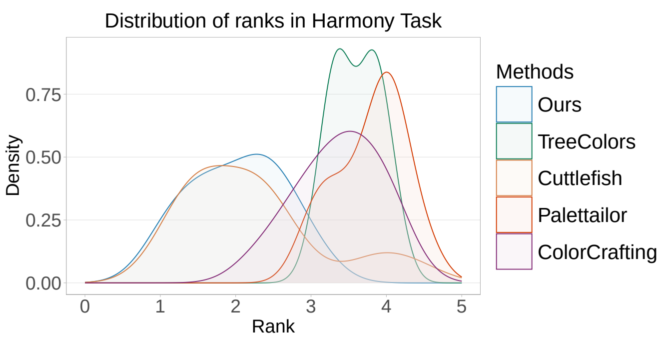


Fig. S7: The distribution of the ranks of different methods on the task of harmony.

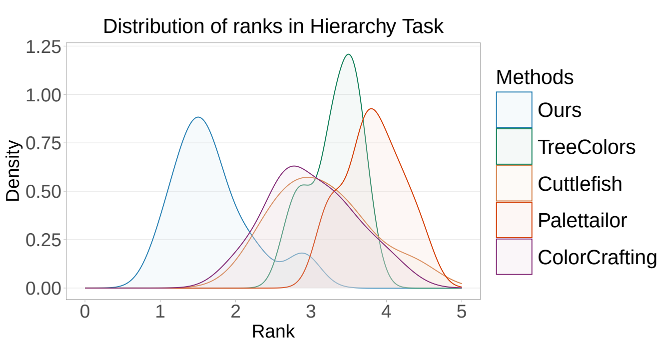


Fig. S8: The distribution of the ranks of different methods on the task of alignment with hierarchy.

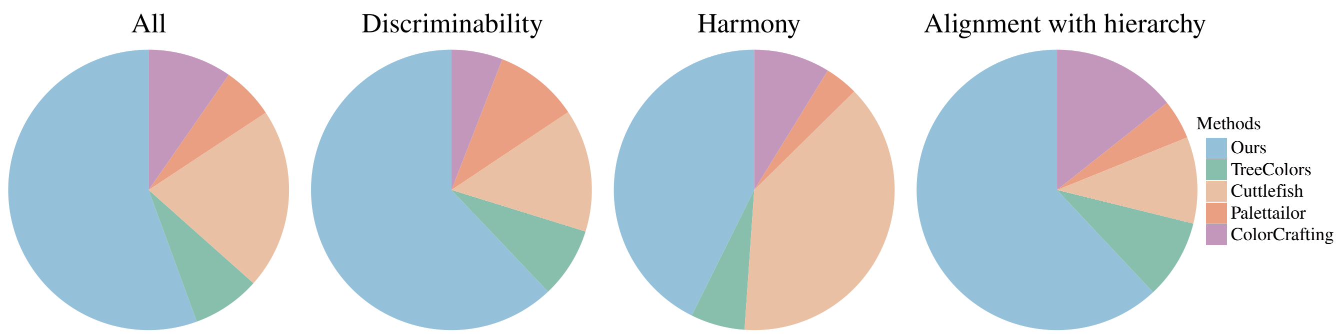


Fig. S9: The frequency of the methods that are ranked first under all the three exploration scenarios.

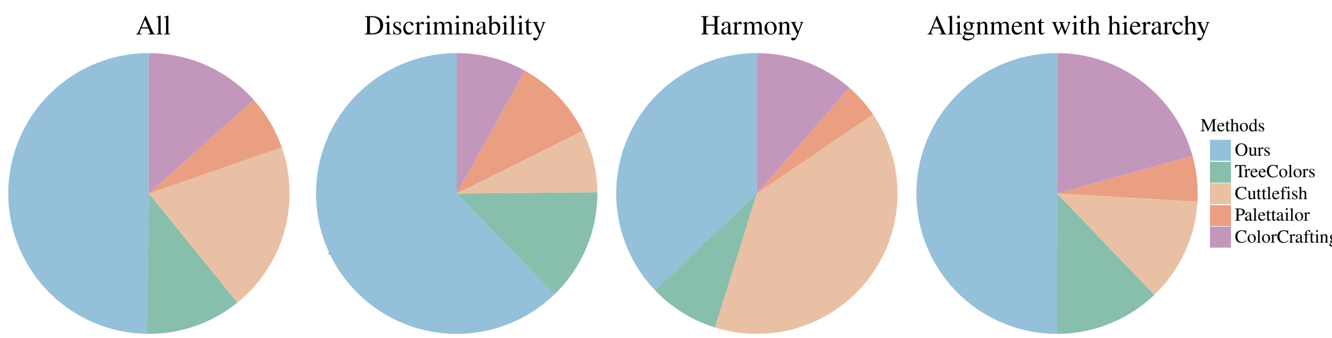


Fig. S10: The frequency of the methods that are ranked first under the exploration scenario that users examine a higher hierarchical level with a balanced subclass distribution.

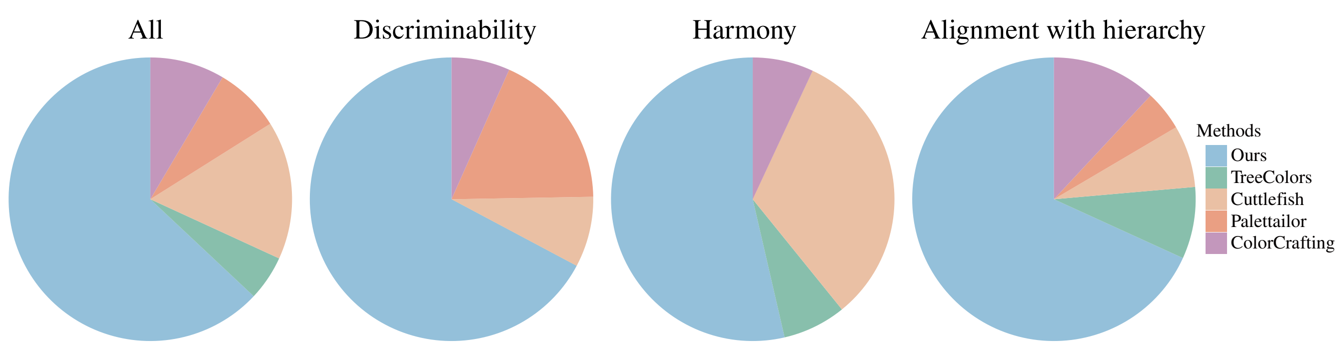


Fig. S11: The frequency of the methods that are ranked first under the exploration scenario that users examine a higher hierarchical level with an imbalanced subclass distribution.

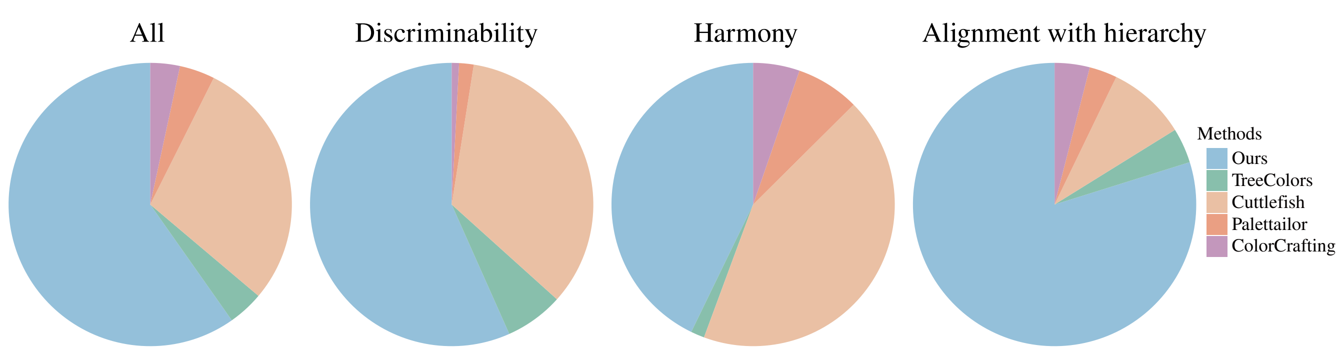


Fig. S12: The frequency of the methods that are ranked first under the exploration scenario that users examine a lower hierarchical level, where the available color range will become much narrower.

REFERENCES

[1] D. Arthur, Asuncion. UCI machine learning repository. https://sci2s.ugr.es/keel/dataset.php?cod=72, 2007. Last accessed 2024-8-1. 1

[2] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *Proceedings of the European Conference on Computer Vision*, pp. 446–461. Springer International Publishing, Cham, 2014. doi: 10.1007/978-3-319-10599-4_29 1

[3] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu. OoDAnalyzer: Interactive analysis of out-of-distribution samples. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3335–3349, 2021. doi: 10.1109/TVCG.2020.2973258 1

[4] A. Corrado. Animals-10 dataset. https://www.kaggle.com/datasets/alessiocorrado99/animals10/, 2019. Last accessed 2024-8-1. 1

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, Miami, 2009. doi: 10.1109/CVPR.2009.5206848 1

[6] H. Fang, S. Walton, E. Delahaye, J. Harris, D. Storchak, and M. Chen. Categorical colormap optimization with visualization case studies. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):871–880, 2017. doi: 10.1109/TVCG.2016.2599214 1

[7] M. Fanty and R. Cole. Spoken letter recognition. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 220–226. Morgan Kaufmann, Denver, 1990. 1

[8] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Visualization*, pp. 43–508. IEEE, San Francisco, 1999. doi: 10.1109/VISUAL.1999.809866 1

[9] G. Griffin, A. Holub, and P. Perona. *Caltech-256 Object Category Dataset*. California Institute of Technology, 2007. 1

[10] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision workshops*, pp. 554–561. IEEE, Sydney, 2013. doi: 10.1109/ICCVW.2013.77 1

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791 1

[12] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang. Palettailor: Discriminable colorization for categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):475–484, 2021. doi: 10.1109/TVCG.2020.3030406 1, 3

[13] Mehta, Anshul. Indian food. https://www.kaggle.com/datasets/anshulmehtakaggl/themassiveindianfooddataset, 2020. Last accessed 2024-8-1. 1

[14] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, Bhubaneswar, 2008. doi: 10.1109/ICVGIP.2008.47 1

[15] G. Sharma and R. Bala. *Digital color imaging handbook*. CRC press, 2017. 3

[16] S. Smart, K. Wu, and D. A. Szafir. Color crafting: Automating the construction of designer quality color ramps. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1215–1225, 2020. doi: 10.1109/TVCG.2019.2934284 1

[17] M. Tennekes and E. de Jonge. Tree colors: Color schemes for tree-structured data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2072–2081, 2014. doi: 10.1109/tvcg.2014.2346277 1

[18] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 595–604. Boston, 2015. doi: 10.1109/CVPR.2015.7298658 1

[19] N. Waldin, M. Waldner, M. Le Muzic, E. Gröller, D. S. Goodsell, L. Autin, A. J. Olson, and I. Viola. Cuttlefish: Color mapping for dynamic multi-scale visualizations. *Computer Graphics Forum*, 38(6):150–164, 2019. doi: 10.1111/cgf.13611 1, 3

[20] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *2019 IEEE Conference on Visual Analytics Science and Technology*, pp. 57–68. IEEE, Vancouver, 2019. doi: 10.1109/VAST47406.2019.8986943 1

[21] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2691–2699. IEEE, Boston, 2015. doi: 10.1109/CVPR.2015.7298885 1