Supplemental Material: Hierarchical Fuzzy-Cluster-Aware Grid Layout for Large-Scale Data

Yuxing Zhou, Changjian Chen, Zhiyang Shen, Jiangning Zhu, Jiashu Chen, Weikai Yang, Shixia Liu

Appendix A: Grid Visualizations in the Requirement Analysis

In Tab. 1, we list all the grid visualizations in the toolkits used by the experts and their disadvantages. In summary, all these grid visualizations are less effective in analyzing fuzzy clusters, and most of them cannot support the analysis of large-scale datasets. This motivated us to develop a new tool, which can support hierarchical exploration and analysis of fuzzy clusters in large-scale datasets.

Appendix B: Dynamic weight adjustment

Since the dynamic weight adjustment is utilized more than once in our method, we give a general example to demonstrate how it works. Suppose that we are optimizing the following function:

$$\min_{\theta} \mu_1 O_1 + \mu_2 O_2 + \mu_3 O_3.$$

Here, O_1, O_2, and O_3 are three objectives to optimize. Weights μ_1, μ_2, and μ_3 balance the three terms and are determined with the multi-task learning method [6]. The key idea of the multi-task learning method is dynamically increasing the weights of the terms that are not well-optimized so that they can be further improved. The degree of optimization is measured by:

$$\Delta(O_i,\theta) = (O_i(\theta) - O_i^{\min})/(O_i^{\max} - O_i^{\min}) \text{ for } i = 1,2,3.$$

$O_i^{\max}$ and $O_i^{\min}$ are the maximum and minimum values of the i-th objective, respectively. Since determining the exact values of $O_i^{\max}$ and $O_i^{\min}$ is challenging, we used two simple strategies to approximate their values. In particular, $O_i^{\min}$ is approximated as the value of the i-th objective when the i-th objective is optimized only, and $O_i^{\max}$ is approximated as the maximum value of the i-th objective when the other ones are optimized:

$$\begin{aligned} O_i^{\min} &= \min_{\theta} O_i, \\ O_i^{\max} &= \max_{j \neq i} O_i(\theta_j) \text{ where } \theta_j = \arg\min_{\theta} O_j. \end{aligned} \tag{1}$$

Initially, μ_1, μ_2, and μ_3 are set as 1 and $\Delta(O_1,\theta)$, $\Delta(O_2,\theta)$, and $\Delta(O_3,\theta)$ are calculated. Then, μ_1, μ_2, and μ_3 are updated:

$$\mu_i = \mu_i + \mathbb{I}_i \cdot 2^{-k}, \text{ for } i = 1,2,3. \tag{2}$$

$\mathbb{I}_i = 1$ if $\Delta(O_i,\theta)$ is the largest among $\Delta(O_1,\theta)$, $\Delta(O_2,\theta)$, and $\Delta(O_3,\theta)$. $\mathbb{I}_i = 0$ if $\Delta(O_i,\theta)$ is the middle value, and $\Delta(O_i,\theta) = -1$ if $\Delta(O_i,\theta)$ is the smallest. k is the current iteration number. Once μ_1, μ_2, and μ_3 are updated, this process is repeated again until the layout process converges.

Appendix C: Simulated Zoom-in Operations in the Quantitative Experiments

In the quantitative experiments, the three baselines (Zhou *et al*.'s method [8], DendroMap [5], and LAS [4]) and our method use different zoom-in operations.

Zhou *et al*.'s method, LAS, and our method. For Zhou *et al*.'s method, LAS, and our method, as users can freely select areas of any positions and sizes in the grid layouts for zooming, we use the same zoom-in operations for them. Specifically, we simulated several zoom-in operations by randomly selecting areas in the grid layouts (*e.g.*, Fig. 1(a)).

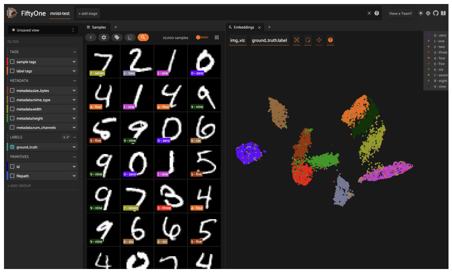
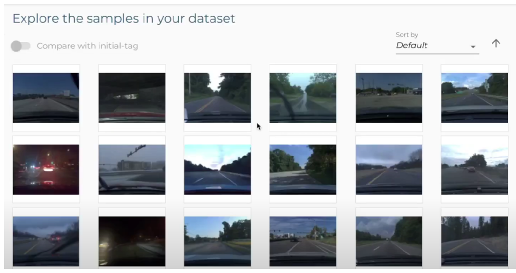
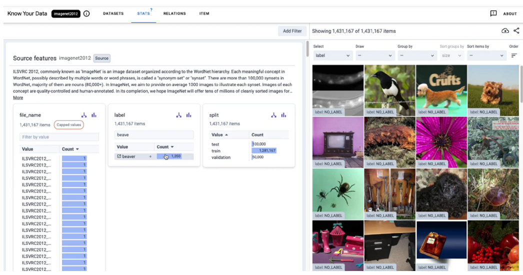
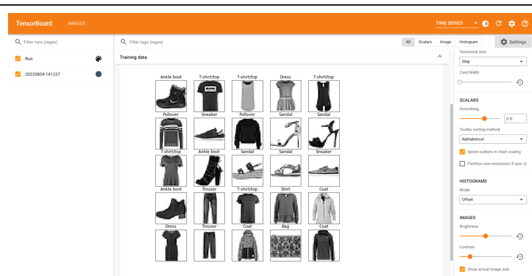
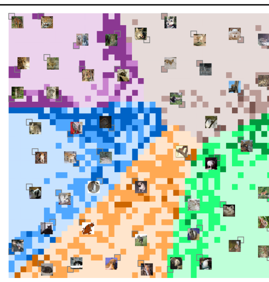
DendroMap. For DendroMap, all samples are organized into a static binary hierarchy. At each time, several nodes in the binary hierarchy are displayed as a treemap. During the exploration, users can only select one treemap node for zoom-in. Fig. 1(b) shows an example of such a zoom-in operation.

Appendix D: Full Quantitative Experiment Results

Experiment detials. For each dataset, we evaluate the baselines and our method with different grid sizes. For the comparison with Zhou *et al*.'s method and LAS, we use sizes of 30×30, 40×40, and 50×50 for experiments. For the comparison with DendroMap, its grid sizes cannot be specified directly. Instead, only the sizes of the images can be specified to change the grid sizes. Therefore, we set the sizes of the images as 30px, 20px, and 16px in a display with a resolution of around 1700×1060. These three image sizes correspond to grid sizes of around 30×30, 40×40, and 50×50.

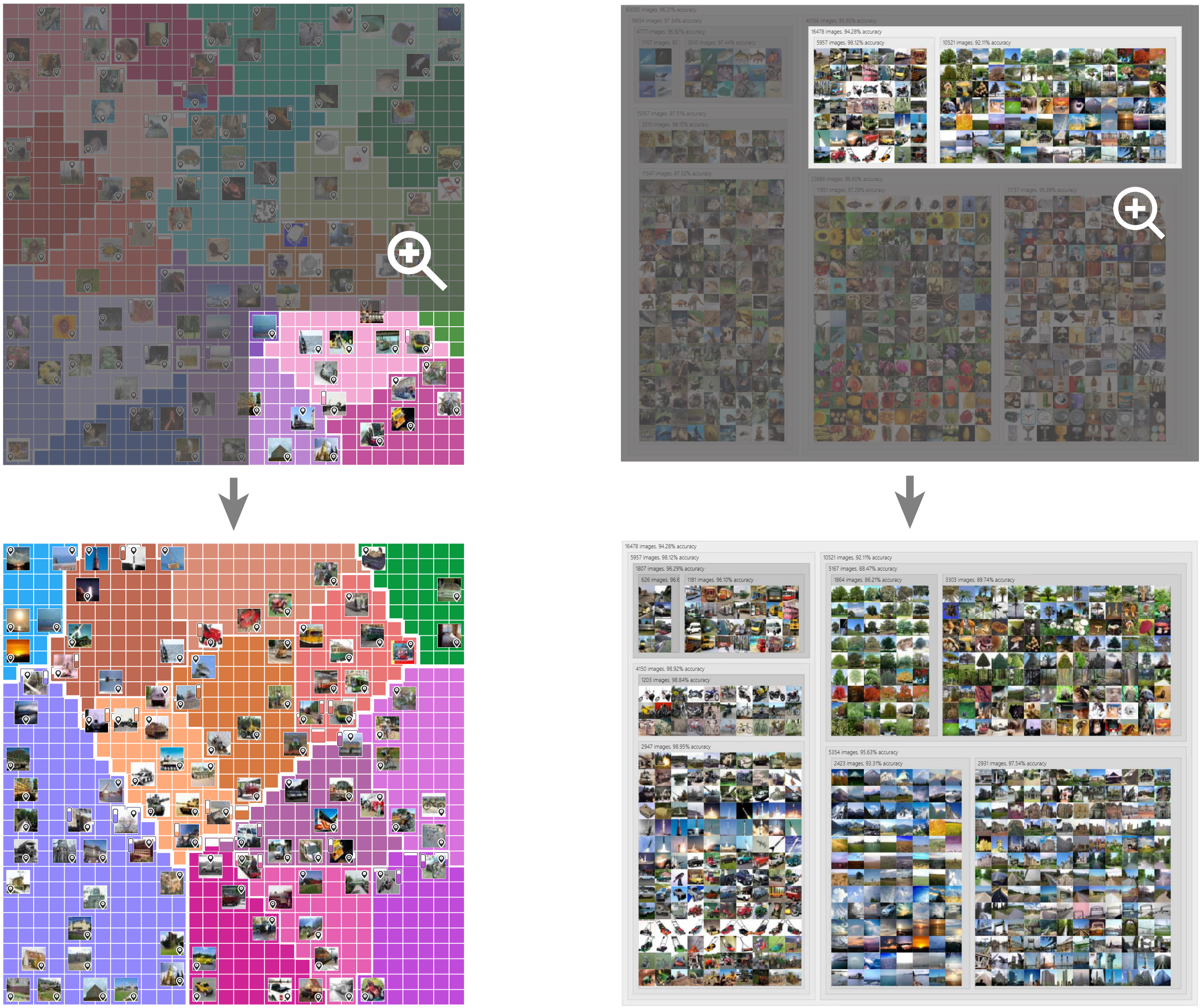
Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Table 1: The grid visualizations in the toolkits used by the experts.

Tool names	Grid visualizations	large-scale datasets	fuzzy clusters
FiftyOne [7]		✗	✗
Lightly [2]		✗	✗
Know Your Data [1]		✗	✗
TensorBoard [3]		✗	✗
Zhou's method [8]		✓	✗

For the comparison between our method, Zhou *et al.*'s method, and LAS, we simulated 150 zoom-in operations for each dataset and each grid size in both methods. For the comparison between our method and DendroMap, as the zoom-in operations are restricted to the treemap nodes, the number of total possible zoom-in operations is limited. Therefore, we first simulated 30 zoom-in operations for DendroMap. Then, for each zoom-in operation, we randomly select five areas with the same grid size in our method. As a result, a total of 150 grid layouts were generated for our method.

Results. For the comparison between our method and Zhou *et al.*'s method, Tabs. 2 to 4 show the comparison of compactness, convexity, proximity, stability, and ambiguity on 3 datasets. For the comparison between our method and LAS, Tabs. 5 to 7 show the results of both methods on 3 datasets. For the comparison between our method and DendroMap, Tabs. 8 to 10 show the results of both methods on 3 datasets.



(a) (b)

Fig. 1: Zoom-in operation examples of (a) Zhou *et al.*'s method, LAS, and our method; (b) DendroMap. Note that each color in (a) corresponds to a cluster in the input hierarchy of the dataset, while a treemap node (rectangle) in (b) corresponds to a node in the static binary hierarchy organized by DendroMap.

Table 2: Comparison of Zhou *et al.*'s method and ours in CIFAR-100. ↑ (↓) indicates the higher (lower) is better.

Conv. type	Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
Triple ratio	30×30	Zhou *et al.*'s method	**0.021**	**0.997**	598.3	0.73	0.092	181.2	1.16
		Ours	**0.021**	0.996	**618.0**	**0.92**	**0.063**	**0.3**	**0.97**
	40×40	Zhou *et al.*'s method	**0.016**	**0.997**	527.6	0.77	0.078	563.9	1.98
		Ours	**0.016**	**0.997**	**566.9**	**0.94**	**0.061**	**0.7**	**1.61**
	50×50	Zhou *et al.*'s method	**0.016**	**0.998**	456.9	0.76	0.088	1475.7	3.30
		Ours	**0.016**	**0.998**	**518.9**	**0.95**	**0.052**	**1.7**	**2.66**
Perimeter ratio	30×30	Zhou *et al.*'s method	**0.022**	0.930	585.9	0.64	0.121	196.0	1.23
		Ours	0.023	**0.965**	**624.0**	**0.89**	**0.081**	**0.7**	**0.94**
	40×40	Zhou *et al.*'s method	**0.018**	0.928	513.1	0.68	0.097	607.0	2.04
		Ours	**0.018**	**0.964**	**562.9**	**0.90**	**0.075**	**1.2**	**1.56**
	50×50	Zhou *et al.*'s method	**0.017**	0.927	449.5	0.65	0.105	1557.9	3.28
		Ours	**0.017**	**0.970**	**515.0**	**0.91**	**0.074**	**3.4**	**2.80**

Table 3: Comparison of Zhou *et al.*'s method and ours in iNat2021-mini. ↑ (↓) indicates the higher (lower) is better.

Conv. type	Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
Triple ratio	30×30	Zhou *et al.*'s method	**0.025**	**0.997**	385.7	0.72	0.225	147.9	5.27
		Ours	**0.025**	0.996	**432.0**	**0.93**	**0.144**	**1.2**	**3.71**
	40×40	Zhou *et al.*'s method	0.020	0.996	364.9	0.61	0.282	432.4	9.14
		Ours	**0.019**	**0.997**	**414.3**	**0.94**	**0.144**	**4.8**	**6.56**
	50×50	Zhou *et al.*'s method	0.020	0.996	350.5	0.63	0.241	1159.8	15.23
		Ours	**0.018**	**0.998**	**415.7**	**0.94**	**0.139**	**14.5**	**10.10**
Perimeter ratio	30×30	Zhou *et al.*'s method	0.033	0.920	369.1	0.58	0.261	163.3	6.74
		Ours	**0.029**	**0.965**	**420.0**	**0.92**	**0.146**	**1.3**	**3.71**
	40×40	Zhou *et al.*'s method	0.026	0.914	345.6	0.51	0.288	453.5	11.00
		Ours	**0.022**	**0.965**	**410.6**	**0.90**	**0.154**	**5.2**	**6.68**
	50×50	Zhou *et al.*'s method	0.028	0.897	315.5	0.51	0.269	1127.0	20.95
		Ours	**0.022**	**0.966**	**413.2**	**0.92**	**0.147**	**12.8**	**11.23**

Table 4: Comparison of Zhou *et al.*'s method and ours in ImageNet-1k. ↑ (↓) indicates the higher (lower) is better.

Conv. type	Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
Triple ratio	30×30	Zhou *et al.*'s method	0.024	**0.996**	539.3	0.61	0.268	183.7	3.47
		Ours	**0.021**	**0.996**	**565.2**	**0.92**	**0.104**	**0.7**	**2.64**
	40×40	Zhou *et al.*'s method	0.018	0.995	489.4	0.51	0.320	556.9	5.84
		Ours	**0.017**	**0.997**	**530.8**	**0.93**	**0.094**	**0.9**	**4.31**
	50×50	Zhou *et al.*'s method	0.019	0.995	446.3	0.48	0.336	1561.4	9.13
		Ours	**0.016**	**0.998**	**492.2**	**0.93**	**0.099**	**3.7**	**7.18**
Perimeter ratio	30×30	Zhou *et al.*'s method	0.030	0.924	516.5	0.50	0.284	196.8	3.41
		Ours	**0.027**	**0.961**	**553.1**	**0.88**	**0.126**	**0.9**	**2.53**
	40×40	Zhou *et al.*'s method	0.024	0.915	467.0	0.43	0.323	589.9	6.42
		Ours	**0.020**	**0.965**	**526.4**	**0.88**	**0.114**	**2.1**	**4.47**
	50×50	Zhou *et al.*'s method	0.024	0.905	415.9	0.43	0.316	1436.7	12.77
		Ours	**0.020**	**0.966**	**491.0**	**0.90**	**0.118**	**5.4**	**7.16**

Table 5: Comparison of LAS and ours in CIFAR-100. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30×30	LAS	0.039	0.777	646.9	0.40	0.651	398.6	5.78
	Ours	**0.021**	**0.996**	618.0	**0.92**	**0.063**	**0.3**	**0.97**
	Ours with proximity only	0.041	0.767	**663.4**	0.39	0.506	379.8	8.15
40×40	LAS	0.029	0.801	576.7	0.40	0.606	1209.2	9.49
	Ours	**0.016**	**0.997**	566.9	**0.94**	**0.061**	**0.7**	**1.61**
	Ours with proximity only	0.033	0.763	**597.6**	0.36	0.551	1229.2	11.72
50×50	LAS	0.028	0.795	528.6	0.38	0.654	3274.7	14.17
	Ours	**0.016**	**0.998**	518.9	**0.95**	**0.052**	**1.7**	**2.66**
	Ours with proximity only	0.034	0.734	**541.9**	0.35	0.552	3232.9	22.06

Table 6: Comparison of LAS and ours in iNat2021-mini. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30×30	LAS	0.092	0.445	**504.3**	0.32	0.491	381.5	63.21
	Ours	**0.025**	**0.996**	432.0	**0.93**	**0.144**	**1.2**	**3.71**
	Ours with proximity only	0.087	0.465	482.5	0.33	0.396	344.5	59.79
40×40	LAS	0.085	0.406	**489.8**	0.28	0.455	1045.4	111.97
	Ours	**0.019**	**0.997**	414.3	**0.94**	**0.144**	**4.8**	**6.56**
	Ours with proximity only	0.078	0.416	464.1	0.29	0.433	1046.0	99.72
50×50	LAS	0.090	0.383	**467.2**	0.25	0.476	2928.0	197.73
	Ours	**0.018**	**0.998**	415.7	**0.94**	**0.139**	**14.5**	**10.10**
	Ours with proximity only	0.080	0.401	441.2	0.27	0.433	2827.1	169.30

Table 7: Comparison of LAS and ours in ImageNet-1k. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30×30	LAS	0.060	0.619	597.9	0.35	0.580	307.6	24.42
	Ours	**0.021**	**0.996**	565.2	**0.92**	**0.104**	**0.7**	**2.64**
	Ours with proximity only	0.062	0.606	**603.8**	0.36	0.437	284.4	27.62
40×40	LAS	0.051	0.604	564.6	0.33	0.541	938.4	35.78
	Ours	**0.017**	**0.997**	530.8	**0.93**	**0.094**	**0.9**	**4.31**
	Ours with proximity only	0.055	0.580	**575.5**	0.33	0.501	961.9	42.80
50×50	LAS	0.050	0.601	530.9	0.32	0.590	2591.9	58.28
	Ours	**0.016**	**0.998**	492.2	**0.93**	**0.099**	**3.7**	**7.18**
	Ours with proximity only	0.055	0.570	**542.2**	0.31	0.497	2594.7	75.64

Table 8: Comparison of DendroMap and ours in CIFAR-100. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30px	DendroMap	0.218	0.624	553.8	0.29	0.334	814.9	2.09
	Ours	**0.051**	**0.943**	**681.0**	**0.85**	**0.080**	**5.6**	**0.59**
20px	DendroMap	0.055	0.675	404.8	0.34	0.355	2054.1	4.62
	Ours	**0.023**	**0.954**	**613.7**	**0.88**	**0.064**	**8.4**	**2.00**
16px	DendroMap	0.060	0.600	344.6	0.33	0.317	5950.5	9.90
	Ours	**0.017**	**0.959**	**596.1**	**0.90**	**0.063**	**12.9**	**3.11**

Table 9: Comparison of DendroMap and ours in iNat2021-mini. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30px	DendroMap	0.136	0.430	349.0	0.35	0.270	317.6	7.87
	Ours	**0.039**	**0.959**	**442.1**	**0.90**	**0.144**	**4.2**	**3.49**
20px	DendroMap	0.150	0.376	268.1	0.22	0.287	1116.0	17.19
	Ours	**0.022**	**0.962**	**399.8**	**0.91**	**0.140**	**16.9**	**6.65**
16px	DendroMap	0.143	0.336	233.4	0.28	0.306	1970.5	27.49
	Ours	**0.022**	**0.966**	**405.3**	**0.92**	**0.142**	**23.0**	**9.60**

Table 10: Comparison of DendroMap and ours in ImageNet-1k. ↑ (↓) indicates the higher (lower) is better.

Size	Method	Comp. ↓	Conv. ↑	Prox. ↑	Stab-shape ↑	Stab-position ↓	Stab-sample ↓	Ambi. ↓
30px	DendroMap	0.074	0.643	469.6	0.48	0.324	436.1	4.93
	Ours	**0.040**	**0.960**	**541.4**	**0.85**	**0.117**	**2.0**	**1.84**
20px	DendroMap	0.069	0.583	361.7	0.40	0.368	447.9	12.19
	Ours	**0.029**	**0.964**	**488.2**	**0.89**	**0.111**	**2.4**	**4.73**
16px	DendroMap	0.071	0.562	317.3	0.40	0.348	939.2	15.77
	Ours	**0.022**	**0.966**	**500.0**	**0.92**	**0.117**	**4.5**	**6.27**

References

[1] Know your data. https://github.com/pair-code/knowyourdata. Last accessed 2024-12-18. 2

[2] Lightly. https://github.com/lightly-ai/lightly. Last accessed 2024-12-18. 2

[3] Tensorboard. https://github.com/tensorflow/tensorboard. Last accessed 2024-12-18. 2

[4] K. U. Barthel, N. Hezel, K. Jung, and K. Schall. Improved evaluation and generation of grid layouts using distance preservation quality and linear assignment sorting. *Computer Graphics Forum*, 42(1):261–276, 2023. doi: 10.1111/cgf.14718 1

[5] D. Bertucci, M. M. Hamid, Y. Anand, A. Ruangrotsakun, D. Tabatabai, M. Perez, and M. Kahng. DendroMap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):320–330, 2023. doi: 10.1109/TVCG.2022.3209425 1

[6] S. Liu, Y. Liang, and A. Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence.*, pp. 9977–9978. Honolulu, 2019. doi: 10.1609/aaai.v33i01.33019977 1

[7] B. E. Moore and J. J. Corso. FiftyOne. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020. 2

[8] Y. Zhou, W. Yang, J. Chen, C. Chen, Z. Shen, X. Luo, L. Yu, and S. Liu. Cluster-aware grid layout. *IEEE Transactions on Visualization and Computer Graphics*, 2023. doi: 10.1109/tvcg.2023.3326934 1, 2