A. Implementation Detail

We state the implementation details of training and evaluating models in this section.

Training. We list the hyperparameters we used in pretraining in Table 1. Note that our learning rate schedule during training is different from most vision-language models: we perform a linear warm-up at the start of every generation, but the long-term trend follows cosine annealing decay.

Hyperparameter	Value
optimizer type	AdamW
base learning rate	0.0005
weight decay	0.1
β_1	0.9
β_2	0.98
lr scheduler	Cosine Annealing
warmup step	500 for every generation
image resolution	224
max token number	77

Table 1. Common hyperparameters used for IL-CLIP pre-training.

Zero-shot image classification. We represent each class by its text description. After extracting the image feature from a target image and text features for all class names, the category of the image can be predicted by choosing the class with the maximum cosine similarity score between its text feature and the image feature. We use the same multiple prompt types as in CLIP paper [3], and the final predictions are averaged between prompts.

B. Additional Experiments on Recognition

We supplement the recognition evaluation by doing linear probing and zero-shot image-text retrieval tasks.

Linear probing. In this evaluation, we classify images by training a linear network layer on top of extracted vision features. Following CLIP [3], We train a logistic regression classifier with L-BFGS optimizer. We set the base learning rate to be 0.05 with no weight decay. The results are shown in table 2. From the fact that our model performs equally well with codebook-CLIP and much better than standard CLIP, we claim the vision representation trained by iterated learning is as powerful in recognition as the normally-trained vision representation.

Zero-shot image-text retrieval. We evaluate all models' zero-shot retrieval performance on the test set of three standard benchmarks: MS-COCO [2], Flickr8k [4] and Flickr30k [4]. The performance is shown in table 3. While iterated learning slightly downgrades the performance of Codebook-CLIP, it still maintains a lead over the standard CLIP model. The performance drop is potentially due to the fact that the text representation is under-trained under the iterated learning framework since the language agent is dynamically replaced.

Pretrain	Method	ImageNet1k	CIFAR-100	CIFAR-10	STL-10	VOC2007	Caltech101	Pets	Flowers102	Food101	Mean
CC3M	CLIP [3]	0.43	0.58	0.80	0.89	0.73	0.80	0.54	0.63	0.45	0.65
	Codebook-CLIP [1]	0.47	**0.62**	**0.84**	0.90	0.74	**0.82**	0.54	0.64	**0.52**	**0.68**
	NegCLIP [5]	0.39	0.56	0.79	0.85	0.72	0.84	0.46	0.55	0.40	0.62
	IL-CLIP (Ours)	**0.49**	0.57	0.80	**0.93**	**0.76**	**0.82**	**0.57**	**0.66**	0.49	**0.68**
CC12M	CLIP [3]	0.60	0.65	0.85	0.95	0.76	0.83	0.72	0.69	0.64	0.74
	Codebook-CLIP[1]	**0.62**	**0.71**	**0.89**	0.96	**0.80**	0.88	0.76	**0.76**	**0.72**	**0.79**
	NegCLIP [5]	0.59	0.63	0.82	0.95	0.75	0.84	0.64	0.70	0.61	0.72
	IL-CLIP (Ours)	**0.62**	0.67	0.85	**0.97**	0.78	**0.90**	**0.79**	0.74	**0.72**	**0.79**
DataComp	CLIP [3]	0.44	0.66	0.85	0.84	0.79	0.82	0.43	0.60	0.52	0.66
	Codebook-CLIP [1]	**0.47**	**0.69**	**0.89**	0.87	0.79	**0.83**	**0.46**	**0.65**	**0.53**	**0.69**
	NegCLIP [5]	0.41	0.60	0.79	0.84	0.76	0.81	0.44	0.58	0.48	0.63
	IL-CLIP (Ours)	0.45	0.68	0.87	**0.88**	**0.80**	**0.83**	0.45	0.63	**0.53**	0.68

Table 2. **Evaluation on Linear probing for all model variants.**

Pretrain	Method	COCO		Flickr8k		Flickr30k		Mean
		IR	TR	IR	TR	IR	TR	
CC3M	CLIP [3]	0.23	0.28	0.41	0.50	0.39	0.48	0.38
	Codebook-CLIP [1]	**0.28**	**0.35**	**0.47**	**0.57**	**0.46**	**0.57**	**0.44**
	NegCLIP [5]	0.19	0.23	0.35	0.42	0.31	0.38	0.31
	IL-CLIP (Ours)	**0.28**	0.32	0.46	**0.57**	0.42	0.51	0.42
CC12M	CLIP [3]	0.39	0.53	0.60	0.75	0.60	0.73	0.60
	Codebook-CLIP [1]	**0.45**	**0.59**	**0.65**	**0.81**	**0.65**	**0.81**	**0.66**
	NegCLIP [5]	0.36	0.48	0.57	0.69	0.56	0.68	0.56
	IL-CLIP (Ours)	0.44	0.56	0.63	0.77	0.64	0.76	0.63
DataComp	CLIP [3]	0.16	0.21	0.24	0.31	0.23	0.32	0.24
	Codebook-CLIP [1]	**0.20**	**0.25**	**0.26**	**0.36**	**0.26**	**0.35**	**0.28**
	NegCLIP [5]	0.13	0.16	0.23	0.29	0.19	0.28	0.21
	IL-CLIP (Ours)	0.18	0.22	**0.26**	0.31	0.24	0.33	0.26

Table 3. **Zero-shot image/text retrieval.** We report retrieval R@5 scores for in three most commonly used retrieval datasets. *IR* stands for image retrieval, *TR* stands for text retrieval.

C. Pretraining on DataComp dataset

We also pretrain our models on DataComp-10M dataset to ensure our finding is not specific to any pretraining dataset. We report the detailed compositionality and image classification accuracy in table 6 and 7 respectively. Their linear probing and image-text retrieval performance are shown along with other variants of models in table 2 and 3. The noisiness of unfiltered DataComp-10M turns out to influence all models' performance, but the IL-CLIP is still the best model in compositionality and comparable to Codebook-CLIP in recognition, which is consistent with the findings in the main paper.

D. Additional Ablation: Iterated Learning with Hard Negative Mining

Our proposed Iterated learning algorithm augments the CLIP training procedure, while NegCLIP augments the CLIP training objective. In principle, these two approach can work together and potentially result in a better model. As an additional ablation, we study a variant of the CLIP model that

Table 4. **Sampled test cases in compositionality benchmarks and performance comparison.** We found our model exhibits better compositional understanding than standard CLIP in distinguishing compositional hard negatives.

uses both iterated learning and hard negative mining during training. We train it on the CC3M dataset. From the results in Table 5, we observe the combination of iterated learning and negative mining yields a model with the best compositionality performance, but leads to a slight performance drop for recognition.

Models	compositionality	classification	probing	retreival
CLIP	0.28	0.22	0.65	0.38
NegCLIP	0.32	0.21	0.62	0.31
IL-CLIP	0.34	**0.24**	**0.68**	**0.42**
IL-NegCLIP	**0.35**	**0.24**	0.67	0.40

Table 5. **Iterated learning with hard negative mining.** Color notations: The performance of the target model that combines negative mining and iterated learning.

E. A user study: Comparing Codebook Interpretability

To compare the interpretability of the trained codebook between IL-CLIP and normal Codebook-CLIP, we conduct a user study where participant annotates whether randomly picked codes have semantically grounded meanings. Across *50* binary decisions on whether specific codes have a semantic meaning, our *10* users annotated *44* codes (in average) to be interpretable in IL-CLIP versus only *39* for codebook-CLIP.

F. Unbiased Visualization for Trained Codebook (Sorted by Index)

To unbiasedly show the performance of our trained codebook, we present a visualization of the foremost codes, organized in ascending order by their index in Fig. 1 - 2. We find most of the codes achieve good semantic groundings, and some of them are interpretable.

G. Qualitative Result of the Models' Performance in Compositional Understanding

In Table 4, we show some qualitative results, including both image-to-text and text-to-image examples. Due to enhanced compositional understanding, we observe our model does better in relationship understanding and counting.

Dataset	Method	CREPE-systematicity		CREPE-productivity			SugarCrepe			Cola	Winoground	Mean
		atom	compound	replace	swap	negate	add	replace	swap	Txt2Img	Txt2Img	
DataComp	CLIP [3]	0.33	0.36	0.11	0.20	0.10	0.63	0.62	0.57	**0.21**	0.10	0.32
	Codebook-CLIP [1]	**0.34**	0.37	0.12	0.21	0.09	0.64	0.64	0.59	0.20	0.07	0.33
	NegCLIP [5]	0.32	0.36	0.11	**0.24**	**0.12**	0.62	0.63	**0.64**	0.19	0.11	0.33
	IL-CLIP (Ours)	**0.34**	**0.40**	**0.14**	0.23	0.09	**0.66**	**0.66**	0.62	0.18	**0.14**	**0.35**

Table 6. **Evaluation on compositionality benchmarks for models pretrained on DataComp-10M.**

Dataset	Method	ImageNet1k	CIFAR-100	CIFAR-10	STL-10	VOC2007	Caltech101	SUN397	Pets	Flowers102	Food101	ObjectNet	CLEVR	Smallnorb	Resisc45	DMLAB	ImageNet-A	ImageNet-R	IN-sketch	Mean
DataComp	CLIP [3]	0.14	0.31	0.72	0.72	0.32	0.62	0.22	0.12	0.05	0.16	0.15	0.12	0.06	0.12	0.13	**0.02**	0.17	0.09	0.24
	Codebook-CLIP [1]	**0.15**	**0.36**	**0.76**	0.72	0.38	**0.68**	**0.25**	**0.14**	**0.07**	**0.18**	0.18	**0.14**	0.06	0.14	0.13	**0.02**	**0.19**	**0.10**	**0.26**
	NegCLIP [5]	0.12	0.28	0.67	0.69	0.32	0.59	0.21	0.11	0.04	0.14	0.16	0.12	0.07	0.12	0.14	**0.02**	0.13	0.07	0.22
	IL-CLIP (Ours)	0.14	0.33	0.74	**0.74**	**0.42**	0.65	0.24	0.11	0.06	0.16	**0.19**	0.13	**0.08**	**0.15**	**0.15**	**0.02**	0.16	0.09	**0.26**

Table 7. **Evaluation of zero-shot image classification with models pretrained on DataComp-10M.**

References

[1] Yuxiao Chen, Jianbo Yuan, Yu Tian, Shijie Geng, Xinyu Li, Ding Zhou, Dimitris N Metaxas, and Hongxia Yang. Revisiting multimodal representation in contrastive learning: from patch and token embeddings to finite discrete tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15095–15104, 2023. 1, 3

[2] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755. Springer, 2014. 1

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 3

[4] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2: 67–78, 2014. 1

[5] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 1, 3

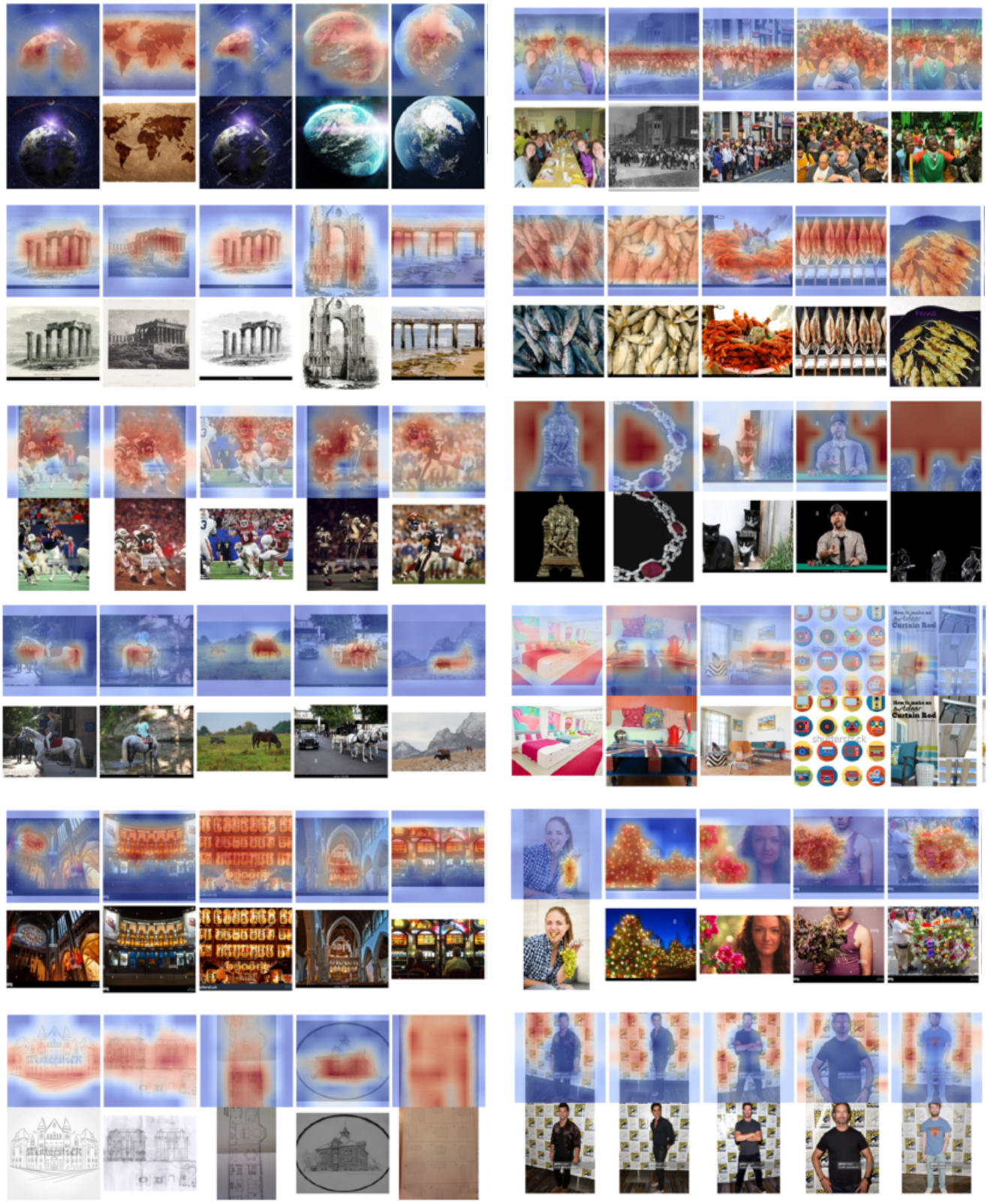

Figure 1. **Codebook visualization:** code #1 - #11

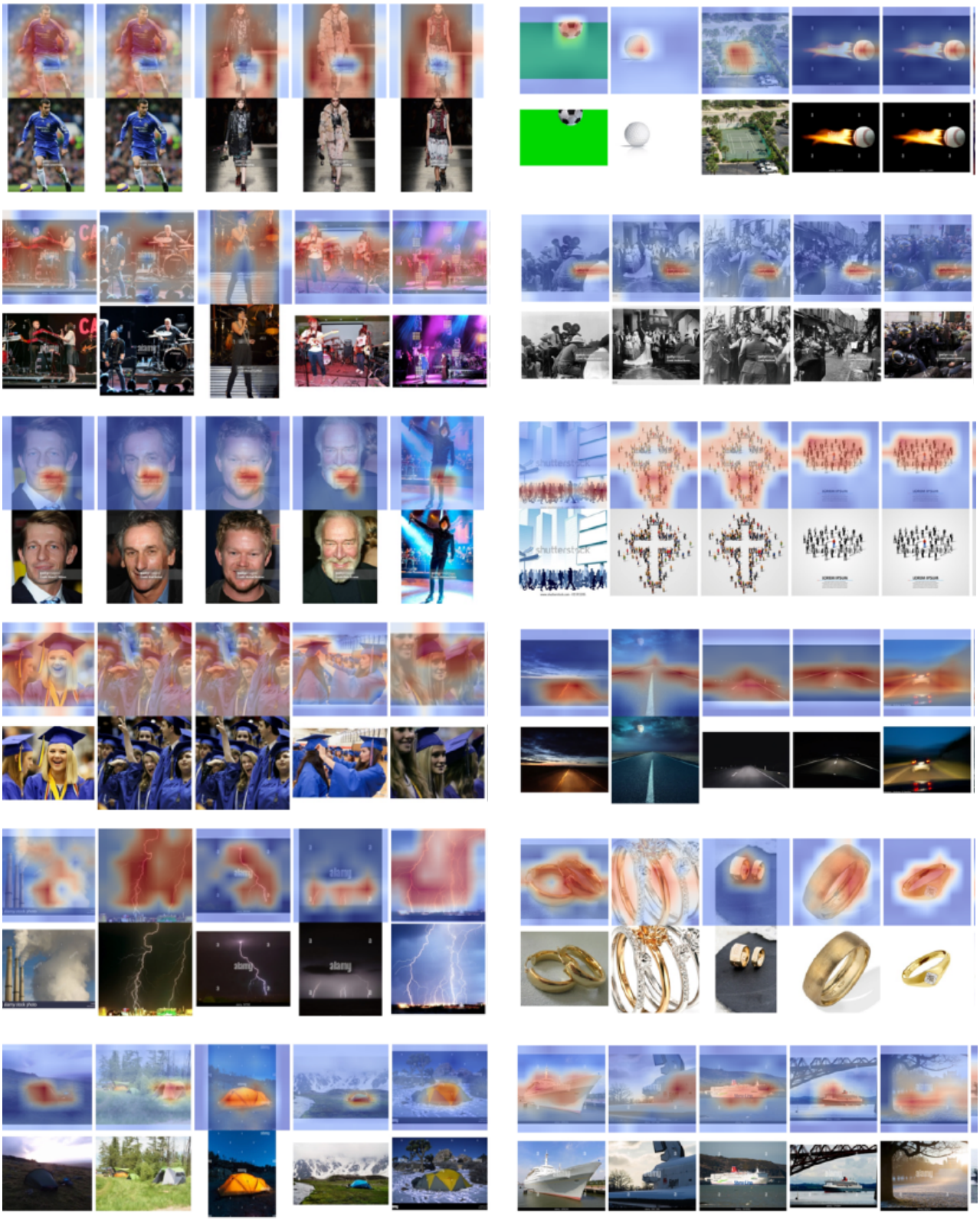

Figure 2. **Codebook visualization:** code #12 - #23