FALCON: False-Negative Aware Learning of Contrastive Negatives in Vision-Language Alignment

Supplementary Material

7. Related Works

Vision-language pretraining (VLP) The training paradigm introduced by ALBEF [30], which jointly optimizes the ITC, ITM, and MLM objectives, has served as the foundation for many subsequent VLP frameworks [5, 7, 8, 17, 21, 22, 31, 32, 51, 54]. Given the high sensitivity of both ITC and ITM objectives to the difficulty of negative samples, several studies have proposed strategies to enhance vision-language pretraining by leveraging hard negative sampling. For example, ALBEF [30] computed the ITM loss by sampling in-batch hard negatives based on contrastive similarity scores computed within the current mini-batch. DiHT [40] proposed an importance sampling approach to upweight harder negatives based on their similarity to the anchor. GRIT-VLP [7] enhances hard negative sampling by introducing the Grouped Mini-batch Sampling (GRIT) strategy, which constructs mini-batches composed of the most semantically similar image-text pairs retrieved from a large candidate pool M. This increases the chance of including informative hard negatives within each batch for both ITC and ITM losses.

False negatives in VLP While several prior works have proposed strategies to mitigate the impact of false negatives in the vision domain [10, 12, 18, 42, 49], the increased risk of false negatives introduced by hard negative sampling remains relatively underexplored in the context of vision-language pretraining. [23] proposed Similarity-Regulated Contrastive Learning (SRCL), which adjusts the contrastive loss by weighting negative samples according to their cross-modal similarity to the anchor, where the similarity is initially estimated using a pretrained model and progressively refined during training. By assigning lower weights to semantically similar negatives, SRCL mitigates the over-penalization of false negatives during contrastive learning. More recently, [6, 8] demonstrated that converting false negatives into positives using a strong pretrained model can improve the performance on downstream tasks. These findings highlight a fundamental trade-off between hard and false negatives, emphasizing its significant impact on the learned representations. However, such methods rely heavily on pretrained models and fixed heuristic thresholds (e.g., ITM score cutoffs) to identify false negatives, which may limit their robustness and generalizability across diverse datasets and training conditions. In contrast, we propose a learning-based approach that adaptively balances the trade-off between hard and false negatives throughout the training process, without relying on fixed heuristics or external pretrained models. [36] proposed a novel geodesic distance metric for multi-modal contrastive learning, designed to more accurately capture the underlying data manifold and thereby better distinguish positive and negative samples. Although this approach is conceptually compatible with FALCON's hard negative scheduling mechanism, we were unable to evaluate its integration due to the absence of an official code release.

Learning to optimize (L2O) L2O is a research paradigm in machine learning that aims to automatically learn optimization algorithms from data, rather than relying on handcrafted update rules. Early works in L2O typically adopted a meta-learning framework, where an optimizer is parameterized (e.g., via neural networks) and trained across a collection of optimization tasks [9, 52, 55]. In this framework, a meta-training set composed of multiple task-specific training and validation dataset pairs is used to guide the optimizer to generalize across tasks. Based on the meta-training set, L2O methods learn parameter update rules that minimize validation loss, either through supervised learning [1, 48] or reinforcement learning [4, 33]. Recent advances have begun to challenge these assumptions by introducing optimization policies that must learn and adapt in the absence of a predefined meta-training set [24]. In this paper, we propose an online optimization approach that constructs mini-batches to balance the tradeoff between hard and false negatives, without relying on any meta-training dataset.

8. Experimental Details

8.1. Experimental Setup

Unless otherwise specified, all experiments follow the training protocols established in [7, 8, 30]. For all retrieval tasks (COCO IRTR [35], Flickr IRTR [38]), we evaluate the pretrained models directly without any task-specific fine-tuning. For NLVR2 [45], we follow the protocol established in ALBEF [30], performing an additional pretraining stage on the COCO dataset to adapt the model for reasoning over paired images, followed by fine-tuning on the NLVR2 dataset for 10 epochs. For the VQA task [2], we fine-tune the pretrained model for 8 epochs using both the training and validation splits of the COCO and Visual Genome datasets [27], following standard practice in prior work [7, 8, 30]. To compute the ITC loss, we employed computationally efficient soft pseudo targets [7, 8] instead of the pseudo targets generated by a momentum model [30] for computational efficiency. For the ITM loss, the negative sample with the highest similarity to the anchor within each mini-batch is selected as the negative

Algorithm 1 Compose Mini-batch Index Set

Input: Similarity matrix $\mathbf{S}$, unselected index set $\mathcal{U}$, batch size B, scheduler π_ϕ

Initialize mini-batch index set $\mathcal{I} = \{\}$
Select quantiles and normalize $\mathbf{S}$ to get $\widehat{\mathbf{S}}$
$q \sim \pi_\phi(\cdot \mid \widehat{\mathbf{S}})$
$i = \text{Uniform}(\mathcal{U})$
$\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
$\mathcal{U} \leftarrow \mathcal{U} \setminus \{i\}$
for $B - 1$ **do**
  $i \leftarrow$ index of q_i-quantile of $\{\mathbf{S}_{i,j} \mid j \in \mathcal{U}\}$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i\}$
end for
return $\mathcal{I}$

Algorithm 2 VLP with Mini-batch Scheduler (for i-th search space M)

Input: VLP parameter θ, scheduler parameter ϕ, Vision dataset $\mathcal{V}$, Text dataset $\mathcal{T}$, learning rate η, γ

Compute pairwise similarity matrix $\mathbf{S}$ between $\mathcal{V}, \mathcal{T}$ in search space M
Initialize unselected index set $\mathcal{U} = \{0, \ldots, |M| - 1\}$
for gradient step $k \in \{0, 1, \ldots, \lfloor |M|/B \rfloor - 1\}$ **do**
  Get mini-batch index $\mathcal{I}$ with Algorithm 1
  Construct mini-batch V, T as samples at indices $\mathcal{I} + i \cdot |M|$ from $\mathcal{V}, \mathcal{T}$
  $\theta_{k+1} = \theta_k - \eta \cdot \nabla_{\theta_k} \mathcal{L}_{\text{VLP}}(V, T; \theta_k)$
  $\Delta_k = \mathcal{L}_{\text{MLM}}(\theta_k) - \mathcal{L}_{\text{MLM}}(\theta_{k+1})$
  $\phi_{k+1} = \phi_k + \gamma \cdot \Delta_k \nabla_{\phi_k} \log \pi_{\phi_k}(q \mid \widehat{\mathbf{S}})$
end for

[7, 8]. For the IRTR task [13], evaluation was performed on the MSCOCO 5K test set. Model training was primarily conducted on a machine equipped with four NVIDIA RTX 4090 GPUs.

8.2. Algorithms

This section presents the pseudo-code for mini-batch construction and the overall training loop of FALCON. Algorithm 1 outlines the procedure for constructing a mini-batch index set $\mathcal{I}$. The process begins by sampling an initial anchor from the pool of unselected index set $\mathcal{U}$, followed by selecting the remaining $B - 1$ indices based on quantile values q drawn from the scheduler π_ϕ. Algorithm 2 describes the overall vision-language pretraining loop with a search space of size $|M|$. The image-text similarity matrix $\mathbf{S}$ is computed from the `[CLS]` embeddings in the current queue. Subsequently, a mini-batch is constructed using Algorithm 1. The vision-language model parameters θ are then updated via gradient descent, while the scheduler parameters ϕ are updated through gradient ascent.

8.3. Implementation Details of the Mini-batch Construction Process

At the beginning of vision-language pretraining, the queue does not yet contain a sufficient number of cached `[CLS]` embeddings to construct the similarity matrix $\mathbf{S}$. Accordingly, we follow GRIT-VLP [7, 8] and adopt a standard uniform mini-batch sampling procedure during the first epoch, without training or applying the scheduler π_ϕ. From the second epoch onward, cached image and text embeddings from the previous epoch are used to compute similarity matrices, enabling the scheduler to guide mini-batch construction. This design introduces a natural warm-start effect, providing a stable and efficient initialization for the learning-based mini-batch sampling scheduler π_ϕ. As training progresses, the cached embeddings are updated epoch by epoch to reflect the current state of the VLP model, enabling the scheduler to make decisions that are aligned with the evolving structure of the embedding space.

To prevent overfitting, we apply instance-level scheduling at the end of each epoch to ensure that the search space does not consist of fixed instances throughout training [7]. This shuffling improves generalization by exposing the scheduler to a more diverse and representative set of training instances over time.

Table 6. Hyperparameter settings used for FALCON

Hyperparameter	**Setting**
Image Resolution	256
Embedding Dimension	256
Batch Size B	96
Masking Probability	0.5
Search Space size $\lvert M \rvert$	{2400, 5664, 28320}
Pretraining Epochs	20
Optimizer	AdamW($\beta = [0.9, 0.999], \lambda = 0.02$)
learning rate γ	scheduled
m for Subsampling	100
Hidden Layer Dimension	256
# Residual Block	2
Optimizer	AdamW($\beta = [0.9, 0.999], \lambda = 0.01$)
learning rate η	1e-4

8.4. Implementation Details of Baseline Methods

For ALBEF [30], GRIT-VLP [7], MAFA [8], and BLIP-2 [32], we conduct experiments using the official codebases released by the original authors.

For methods without publicly available implementations (DiHT [40] and SRCL [23]), we implemented the loss functions described in the respective papers using the ALBEF codebase as a foundation.

For quantile-based heuristic baselines, we adopt the mini-batch grouping procedure of GRIT-VLP, modifying the de-

Table 7. Retrieval performance of two baseline models (GRIT-VLP, MAFA) pretrained on the MSCOCO dataset under various hyperparameter configurations. All other settings are fixed.

Component	Setting	Text Retrieval			Image Retrieval		
		R@1	R@5	R@10	R@1	R@5	R@10
GRIT-VLP	$\|M\| = 1920$	60.60	83.52	89.14	44.61	69.54	77.67
	$\|M\| = 4800$	55.08	78.10	84.60	39.06	62.57	71.67
MAFA	$\|M\| = 1920, \tau = 0.98$	60.96	83.24	89.62	44.77	69.49	77.96
	$\|M\| = 4800, \tau = 0.98$	54.86	77.04	84.36	39.57	63.13	72.20
	$\|M\| = 1920, \tau = 0.80$	40.62	68.04	78.20	33.10	57.75	67.63

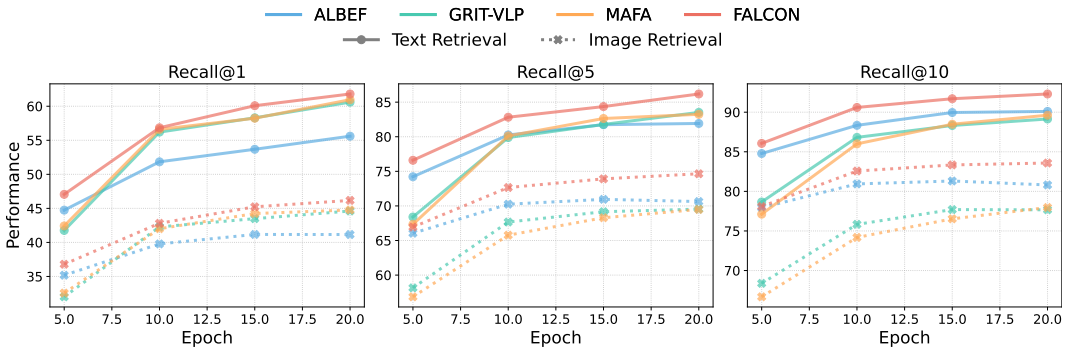

Figure 8. Performance comparison of VLP models across vision-language pretraining epochs on IRTR task. Recall@K (K = 1, 5, 10) is reported separately for text-to-image (solid lines) and image-to-text (dotted lines) retrieval.

fault quantile $q = 1.0$ according to each heuristic strategy. These include fixed quantile settings, such as $q = 0.5$ and $q = 0.0$, as well as dynamic schedules in which the quantile is progressively increased (hardening) or decreased (softening) over the course of training.

For experiments involving SigLIP-2 [47], we initialized our models using the official pretrained checkpoint, as the training code has not been publicly released. We continued pretraining on the MSCOCO dataset using the hyperparameters reported in the original paper for five epochs. To ensure stable training, we excluded the SILC/TIPS loss, which we empirically found to cause instability during continued pretraining. Additionally, we replaced the original sigmoid-based contrastive loss with a softmax contrastive loss to more effectively exploit the benefits of hard negative batching.

8.5. Hyperparameter Settings

We use the same backbone architecture and data augmentation strategy as ALBEF. The detailed hyperparameter settings are summarized in Table 6. For all remaining configurations, we follow the settings used in GRIT-VLP.

8.6. Pretraining dataset size

Table 8 shows the statistics of the pretraining dataset we used. For Conceptual Captions dataset, we used the preprocessed version provided by the original authors of BLIP [31].

Table 8. Statistics of the pretraining dataset

	COCO	VG	CC + SBU
image	113K	100K	3.63M
text	567K	769K	3.63M

9. Additional Experiment results

9.1. Hyperparameter Sweeping in Baselines

To ensure fair and competitive baselines, we sweep the search space size $|M|$ for both methods on the MSCOCO dataset and report in Table 7. For MAFA, we additionally sweep the similarity threshold τ, which determines whether a given image-text pair is classified as a missed-positive (i.e., filtered out from negatives).

9.2. Comparison with Heuristic Negative Mining Methods Across Training Epochs

We visualize the learning curve of FALCON against baseline methods over the full course of epochs on the image-text retrieval (IRTR) downstream task in Figure 8. All models are pretrained on the MSCOCO dataset. Throughout training, FALCON consistently outperforms all heuristic baselines across all epochs and recall metrics, highlighting its

Table 9. Performance comparison of baseline models pretrained on the Conceptual Captions dataset [44], both with (left) and without (right) refinement using the BLIP captioner, in addition to the MSCOCO dataset.

Method	Clean (1.1M pretrain dataset)						Noisy (1.1M pretrain dataset)					
	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
ALBEF	66.38	88.52	93.98	50.77	77.59	86.00	63.92	87.48	93.20	48.53	75.96	84.99
GRIT-VLP	66.42	86.62	91.90	48.90	73.21	80.60	**67.92**	**88.66**	93.26	50.00	75.30	82.65
MAFA	66.36	88.60	92.90	50.70	74.83	83.66	65.94	85.58	90.52	49.39	72.42	79.33
FALCON	**67.34**	**89.06**	**94.26**	**51.81**	**78.78**	**86.67**	66.00	87.90	**93.98**	**50.47**	**77.52**	**86.02**

Table 10. Zero-shot MSCOCO performance comparison of models pretrained on the subset of DataComp dataset.

Method	Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
GRIT-VLP	11.00	27.94	40.14	7.82	22.15	32.62
MAFA	11.24	28.82	40.10	7.67	22.05	31.81
FALCON	**12.88**	**32.42**	**44.30**	**8.76**	**24.24**	**34.81**

effectiveness in adaptively selecting negative samples with appropriate hardness during VLP mini-batch construction.

9.3. Comparison on BLIP-Captioned and Noisy Datasets

To further evaluate the robustness of FALCON in mitigating false negatives, we conducted experiments on web-crawled image-text pairs from the Conceptual Captions dataset [44], with and without refinement using the BLIP captioner [31], in addition to the MSCOCO dataset. This evaluation assess whether FALCON can leverage high-quality captions generated by BLIP to improve performance on noisy, web-crawled data. As shown in Table 9 (left), FALCON significantly outperforms heuristic baselines and demonstrates further performance gains as caption quality increases, compared to the results in Table 1. However, when the same web-crawled data is used without BLIP-based refinement, the performance gains become less pronounced, particularly in the text retrieval task (Table 9 (right)). We attribute this to significant noise and semantic misalignment in the original captions, which can hinder accurate estimation of the tradeoff between hard and false negatives, thereby making the learning process more complex.

9.4. Comparison on DataComp dataset

To demonstrate the generality and robustness of FALCON, we pretrained FALCON and baselines on a 1M subset of 12.8B DataComp [14] filtered by CLIP similarity and English filtering. We evaluate zero-shot performance on the MSCOCO image-text retrieval benchmark, as well as on the full suite of 38 classification and retrieval tasks from the DataComp benchmark. The results are reported in Table 10 and Table 11, respectively.

Table 11. Zero-shot 38 downstream tasks performance comparison of models pretrained on the subset of DataComp dataset.

Dataset	GRIT-VLP	MAFA	FALCON
Caltech-101	28.9	33.9	35.3
CIFAR-10	50.6	71.2	69.0
CIFAR-100	10.1	23.6	25.4
CLEVR Counts	13.4	12.6	13.5
CLEVR Distance	24.6	22.2	20.9
Country211	1.5	2.0	2.4
Describable Textures	11.3	17.2	15.4
EuroSAT	21.0	16.4	22.8
FGVC Aircraft	1.6	1.2	1.6
Food-101	4.9	8.2	3.9
GTSRB	3.7	5.8	4.0
ImageNet 1k	7.4	11.9	14.5
ImageNet Sketch	2.7	4.6	5.9
ImageNet v2	6.4	9.9	12.1
ImageNet-A	3.3	5.0	5.8
ImageNet-O	13.5	18.0	22.0
ImageNet-R	5.7	7.1	8.1
KITTI Vehicle Distance	11.3	10.0	10.9
MNIST	7.4	9.1	10.5
ObjectNet	2.5	3.4	3.5
Oxford Flowers-102	2.7	1.7	1.7
Oxford-IIIT Pet	7.4	10.0	7.7
Pascal VOC 2007	33.0	35.4	39.4
PatchCamelyon	51.3	52.0	50.8
Rendered SST2	50.0	49.9	50.1
RESISC45	16.2	13.7	15.3
Stanford Cars	2.3	3.6	3.3
STL-10	75.8	81.9	83.3
SUN397	9.3	11.2	13.2
SVHN	7.8	7.3	7.3
Flickr	9.9	12.7	14.9
MSCOCO	5.3	7.1	7.7
WinoGAViL	37.7	37.4	40.2
iWildCam	0.5	0.7	1.6
Camelyon17	51.4	50.9	51.7
FMoW	0.0	1.9	0.0
Dollar Street	33.9	34.9	38.8
GeoDE	33.5	48.0	45.6
Average	18.1	20.7	**21.3**

For comparison under a matched data budget, Table 9 (left) presents results on the filtered Conceptual Captions

Table 12. Performance Comparison of FALCON with baselines under the BLIP-2 framework.

Method	Stage-1 COCO Text Retrieval R@1	R@5	R@10	COCO Image Retrieval R1	R@5	R@10	Flickr R@1 TR	IR	Stage-2 VQA2 val	OKVQA test	GQA test-dev	Captioning CIDEr	SPICE
BLIP-2	75.22	93.00	96.50	57.98	82.08	88.78	90.10	77.48	42.46	17.94	28.87	**107.2**	**19.5**
+ GRIT-VLP	73.90	93.10	96.52	57.47	80.50	87.56	90.40	77.28	39.91	15.92	27.75	105.9	19.4
+ MAFA	74.21	93.00	96.61	57.94	81.12	88.44	90.30	77.32	41.12	18.54	28.93	106.1	19.3
+ FALCON	**75.56**	**93.50**	**96.90**	**58.52**	**82.39**	**88.98**	**90.90**	**77.72**	**42.67**	**20.96**	**29.29**	106.0	19.4

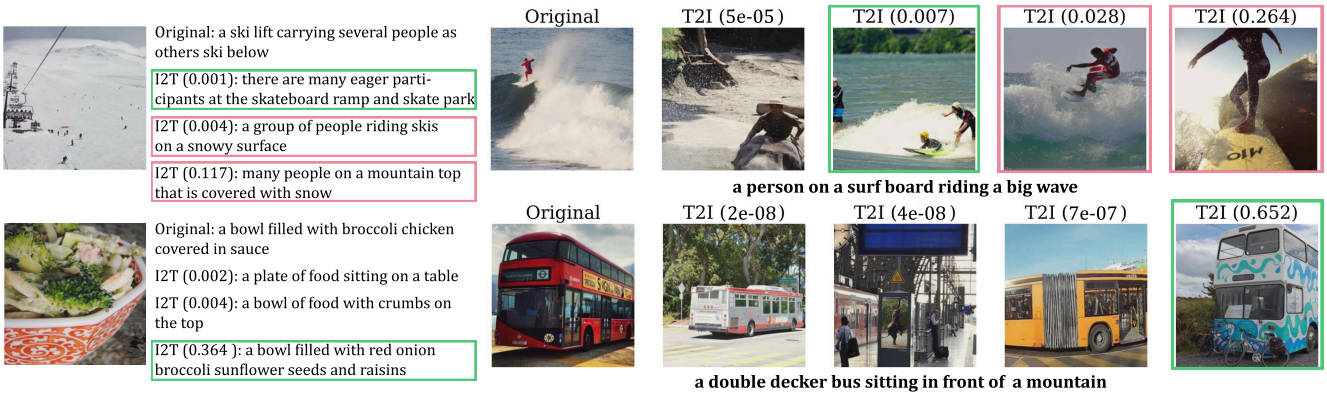

Figure 9. Image-to-Text **(Left)** and Text-to-Image **(Right)** examples of negative sampling under FALCON's quantile-based scheduling strategy. Negative candidates are drawn from similarity score quantiles [0.8, 0.9, 1.0] for I2T and [0.5, 0.8, 0.9, 1.0] for T2I. The negative sample selected by FALCON is highlighted in green and the genuine false negative sample is highlighted in red.

dataset of approximately 1.1M pairs—comparable in scale to the 1M DataComp subset. Under this controlled setting, FALCON achieves an average improvement of 2.4% over MAFA in Table 9, while this margin increases to 11.9% in Table 10.

We attribute this discrepancy to a key limitation of MAFA, its reliance on a fixed pretrained model (BLIP-129M) for false-negative detection. This reliance inherently couples its effectiveness to the domain and distributional characteristics of the data used during the pretraining of the filtering model. In our experiments, MAFA's performance degrades significantly when applied to DataComp dataset, which deviates significantly from the BLIP-129M model's original pretraining corpus. These findings underscore the need for adaptive methods like FALCON, which can dynamically identify false negatives based on the current training data, regardless of prior pretraining exposure.

10. Limitations and Future work

As discussed in Section 4.3, our findings suggest that for FALCON to be fully effective, the proxy signal used for cross-modal alignment should integrate information from both the vision and text encoders. A promising future direction is to develop learning-based strategies for scheduling the trade-off between hard and false negatives in vision–language pretraining that do not rely on such auxiliary objectives, thereby enabling broader applicability across contrastive learning paradigms [41].

Furthermore, the recent emergence of Large Vision-Language Models (LVLMs) has demonstrated strong performance across a wide range of multimodal tasks. These models are typically built upon VLP backbones that provide the core cross-modal representations [3, 29, 50]. We believe that continued improvements in these VLP backbones, along with advances in contrastive learning (e.g., false-negative-aware strategies like FALCON), will contribute to the future development and effectiveness of LVLMs.

Table 13. Performance on downstream vision-and-language tasks.

Method	#Images	VQA test-dev	test-std	NLVR^2 dev	test-P
UNITER [11]	4M	72.70	72.91	77.18	77.85
OSCAR [34]	4M	73.16	73.44	78.07	78.36
VILLA [15]	4M	73.59	73.67	78.39	79.30
ViLT [25]	4M	70.94	–	75.24	76.21
ALBEF [30]	4M	74.54	74.70	80.24	80.50
TCL [51]	4M	74.90	74.92	80.54	81.33
GRIT-VLP [7]	4M	75.11	75.26	80.73	81.60
MAFA [8]	4M	75.55	75.75	82.52	82.08
FALCON	4M	**75.62**	**75.78**	**82.61**	**82.28**

11. Additional Visualization of FALCON

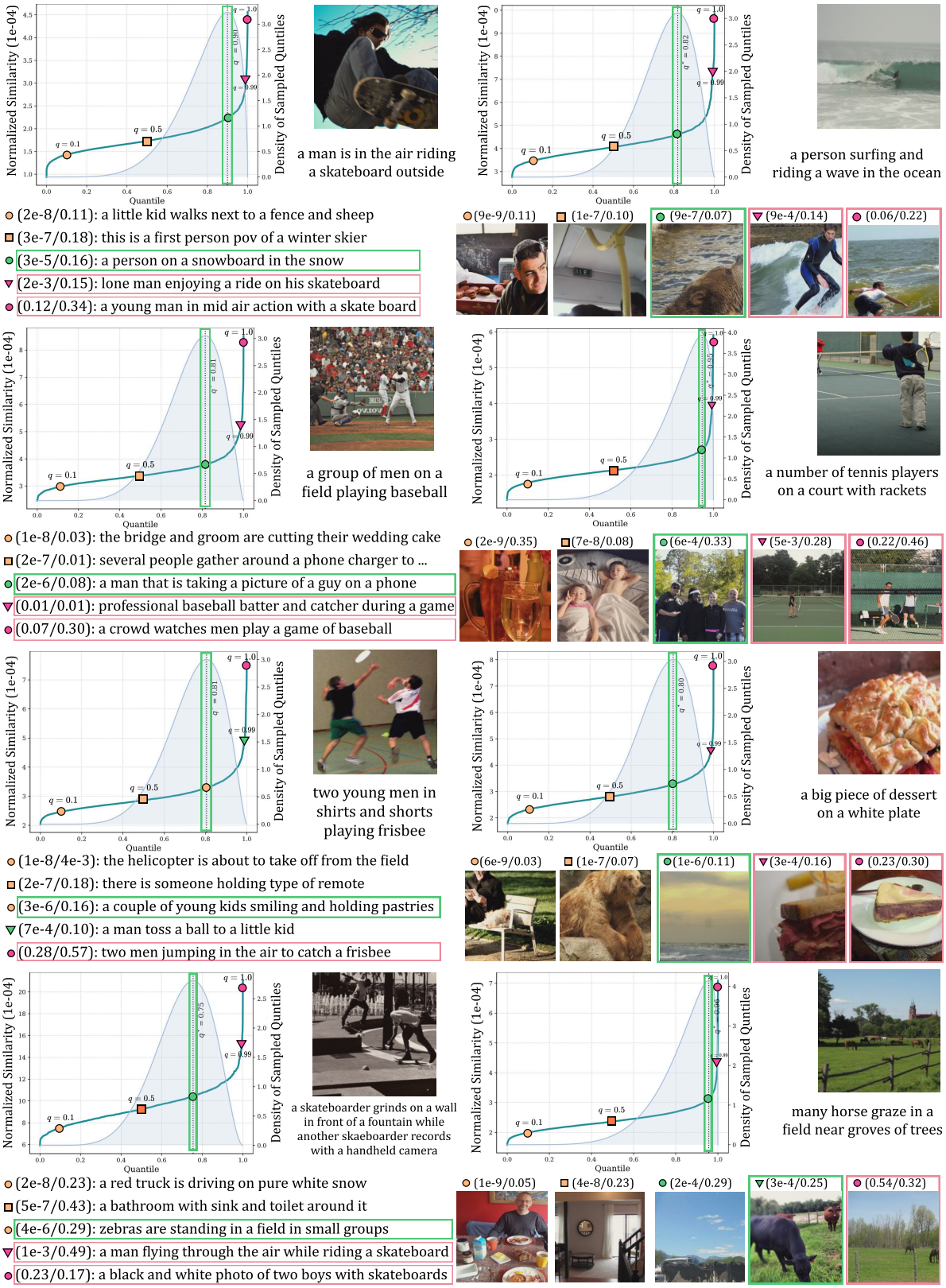

Figure 10. Additional anchor-specific negative sampling visualizations. We highlight the mode of scheduler distribution in green and genuine false negatives in red. Each negative is annotated with "(one-way similarity / ITM score)" and its hardness is color-coded as in Figure 1.