Appendix for Cross-Domain Image Captioning with Discriminative Finetuning

A. Crowdsourcing Experiment Details

We chose the stimuli for the human annotation experiment as follows. We iterated over the images in our Conceptual Captions test set, and sampled, for each image, the 9 closest neighbours, thus creating sets composed of 10 images: 1 target and 9 distractors. We set a threshold of maximum cosine similarity to the nearest neighbour of 0.8. We decided on this threshold after visual inspection of the generated sets: we aimed at having sets challenging for the annotators, yet not impossible to solve, and higher thresholds could lead to sets containing almost identical images, such as subsequent frames extracted from the same video or different croppings of the same picture. Neither targets nor distractors were repeated in the sets and we manually excluded disturbing images.

In the human retrieval experiment, we annotated each set with 3 types of captions: human captions, or captions generated by DiscriTune(-ConCap) or ClipCap(-ConCap), respectively. We randomly divided the entire set into blocks of 100 questions containing mixed caption types. On each screen, the 10 images from a set were presented at the center, arranged in two arrays of 5 images, with the caption written above–see Figure 1. Participants were asked to click on the image that matched the caption best. They were shown one example before starting the task, and were also warned that some cases could contain automatically-generated captions: we asked them to always reply with the answer they found most plausible. Finally, they were warned that the experiment contained some control items, used to ensure annotation quality.

Each subject was presented with one block of questions, plus 5 randomly placed controls, designed to ensure that annotators were paying attention to the task. These cases were made intentionally very simple: targets were surrounded by random distractors not appearing in the other sets, and the associated caption was a human generated one. We made sure internally that these sets could be easily processed with 100% retrieval accuracy.

The data collection routine was written in Psychopy [7] and launched through Pavlovia.[1] There was no time limit for completing the study. We recruited participants via

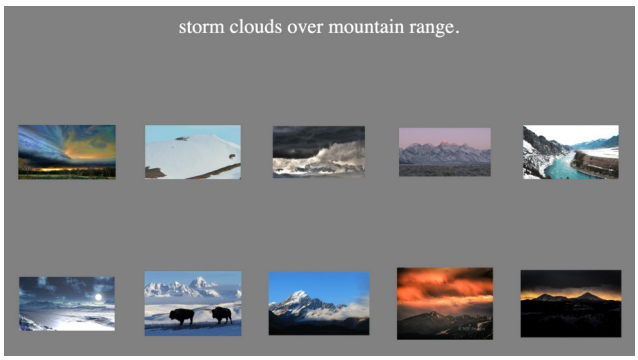

Figure 1. Example of a screen shown to the participants, with a human caption.

Amazon Mechanical Turk.[2] We only accepted annotators from the US, with HIT approval rate higher than 97%, and number of approved HITs higher than 1000. We informed them that we would not collect any personal data (except for their workerID, necessary for their payment, that we would not make public), and that the goal of the experiment was to study how well people identify images based on descriptions. Before being able to access the link of the experiment, participants had to complete an informed consent form, warning them that our experiment would show images and descriptions sampled from the web, and that could therefore contain upsetting content (although, as said above, we manually ensured that images we personally found disturbing would be excluded). They were able to quit the experiment at any time. We paid them 13.5$ for completing the task. The experiment was approved by the ethical board of Universitat Pompeu Fabra in the context of the AMORE project (grant agreement No. 715154). We excluded the data of participants that made more than one mistake when scoring the controls, suggesting that they were not paying enough attention to the task.

B. Hyperparameter Exploration

To generate text, we use greedy decoding at train time and beam size with 5 beams at test time, without tuning these values. hyperparameter searches, we use retrieval score as our selection metric, since its consistent with our

[1] https://pavlovia.org/

[2] https://www.mturk.com

training objective and does not require annotated data. We perform our hyperparameter searches on the Flickr validation set. Even though finetuning on Flickr might be seen as favouring out-of-domain performance (the models were pre-trained on COCO or Conceptual Captions), we we are confident it is not a major factor: the following sections show that the type of REINFORCE baseline and reward function do not have a big impact on performance, and for learning rate we informally found that, as long as large values are avoided, it does not greatly affect results, only convergence speed. Finally, optimizing text-based models with reinforcement learning, especially when done form scratch, can be a challenging problem due to sparse rewards and the vast action space of selecting a token from a vocabulary. This can lead to repetitions or other unnatural word sequences. In practice, we did not notice issues such as repetitions or ungrammatical text. Indeed, we observed quite the opposite, with DiscriTune consistently producing natural text, as confirmed by the NLG metrics improvements. We believe this is due to starting from pre-trained models that are producing fluent language. Evidently, discriminative REINFORCE tuning does not degrade fluency.

B.1. Reinforce Baseline

Using REINFORCE [9], we can rewrite the gradient of the expected reward as the expectation of the gradient, approximated by a single sample caption $\hat{c}$:

$$\begin{aligned}\nabla_\theta L(i, D, \theta) &= \nabla_\theta \mathbb{E}_{c\sim P_\theta(\cdot|i)}[-R(c, i, D)] \\ &= \mathbb{E}_{c\sim P_\theta(\cdot|i)}[-R(c, i, D)\nabla_\theta \log P_\theta(c|i)] \\ &\approx -R(\hat{c}, i, D)\nabla_\theta \log P_\theta(\hat{c}|i)\end{aligned} \tag{1}$$

where i is the target image, $\hat{c}$ is the generated caption, D is the set candidates fed to the retriever and R is the reward function. The parameters θ can be optimized with regular (mini-batch) gradient descent. To reduce variance of the gradient estimator when using REINFORCE, we subtract a baseline term. We compared two different baselines. The first is a running mean of past rewards values using greedy decoding. The second uses beam with the baseline computed using the reward value given by CLIP when fed captions generated with greedy decoding. We trained a ClipCap model on Flickr for 10 epochs using the setup described in Section 4, and then evaluated its performance on the validation set. Results are presented in Table 1. We found that, without subtracting a baseline, the model performed poorly, achieving an accuracy of 34.3%. Running mean and greedy decoding yield similar performance, with running mean showing slightly higher accuracy (97.8% vs 97.4%). We thus employed a running mean baseline with greedy decoding in all the main experiments.

baseline type	P@1
no baseline	34.3
greedy decoding (w/ beam search)	97.4
running mean	97.8

Table 1. ClipCap retrieval accuracy with 100 candidates on the Flickr validation set using different REINFORCE baselines. The *no baseline* and *running mean* methods were used employing greedy decoding to generate captions, whereas when *greedy decoding* was the baseline, we let ClipCap produce captions with beam search using 5 beams.

B.2. Reward Function

To find the best reward to train our captioner, we trained a ClipCap model on Flickr for 10 epochs using the setup described in Section 4, and then evaluated its performance on the validation set. We explored three different reward functions. The cosine similarity reward computes the normalized dot product between the target image embedding and the model-generated caption representation. This is equivalent to the CLIPScore [4] and it is not discriminative since it does not compare the target image with any distractor. The accuracy-based reward computes a binary score which is 1 if CLIP assigned the highest dot-product-based alignment score to the target image when fed a caption, and 0 otherwise. The third reward type is the negative softmax-normalized log probability of the match between a caption and each image in the candidate list, as described in Section 3. As reported in Table 2, the log probability reward performed best, although not by a large margin. Thus, we run all the experiments presented in this work optimizing the captioner using such reward.

reward function	P@1
cosine similarity	85.2
accuracy	85.3
log probability	86.2

Table 2. ClipCap retrieval accuracy with 100 candidates on the Flickr validation set with different reward functions.

C. Finetuning CaMEL

We apply our DiscriTune method to the recently introduced CaMEL [2] captioner model. This model is trained on COCO using a distillation loss based on a model tracking the running mean of an online network, and concurrently optimized with a reward-based objective after a first phase of supervised learning against human references. The reward is computed using CIDEr [8] (please see [2] for additional details on the model and its training setup). NLG results in Table 3 confirm that, at the price of a small drop in in-domain performance, DiscriTune is able to improve (by

COCO				
Model	B@4	M	C	S
CaMEL	**38.11**	**29.03**	**128.62**	**23.35**
DiscriTune-CaMEL	33.45	27.63	117.71	22.03
Flickr				
CaMEL	**22.93**	20.93	58.38	14.62
DiscriTune-CaMEL	22.60	**20.99**	**59.12**	**14.94**

Table 3. NLG metrics (BLEU@4 [6], METEOR [3], CIDEr [8] and SPICE [1]) for CaMEL and DiscriTune-CaMEL captions on the COCO test split (in-domain, our results when using CaMEL) and Flickr test split (out-of-domain).

a small margin) when tested on the Flickr out-of-domain dataset, confirming the benefits of discriminatively finetuning a pre-trained captioner, even when the procedure is applied to a "bleeding-edge" model of this sort.

D. Image Retrieval with Hard Distractors

D.1. ImageCoDe

In order to test retrieval performance in a challenging setup, we use the **ImageCoDe** dataset [5]. ImageCoDe was recently introduced as a testbed for text-based image retrieval. It is formed by 10-elements sets of target images collected from consecutive video frames or by mining similar images to a given target frame. For a fair comparison with prior work, we use the validation images as test data.

In Table 4, we report ImageCoDe results with all our model-generated captions as well as human-generated ones, when a CLIP model with ViT-B-32 was used as the text-to-image retriever. The models are the ones trained with the setup described in Section 4. The results are remarkable, reaching a new state of the art (for either human or model-generated captions) on this dataset (best previous result, obtained with human captions: 29.9% [5]). This shows that our tuning method is beneficial to produce discriminative captions even in contexts in which distinctions need to be very subtle. This suggests that our method could be profitably applied to scenarios where such granular discrimination is called for, such as in video understanding tasks.

D.2. Hard Negative Mining

We perform an additional experiment where at training time the retrieval task is performed using automatically-mined hard distractors. When testing, we still randomly select all non-target candidates. We pick the k most similar distractors based on the cosine similarity with a target using the CLIP visual encoder (the remaining $99 - k$ distractors are picked randomly, as usual). This experiment is aimed at studying the impact of the distractors in discriminative finetuning, with the idea that making the task harder should lead to more discriminative captions. In Table 5 and

Captions	ImageCoDe
ClipCap-COCO	28.7
DiscriTune-COCO	34.0
ClipCap-ConCap	26.8
DiscriTune-ConCap	**36.2**
Blip-COCO	24.0
DiscriTune-COCO	24.7
Human	22.3

Table 4. Percentage accuracy (P@1) when retrieving a target image from the validation image sets of ImageCoDe. Random chance is at 10%.

COCO				
Model	B@4	M	C	S
ClipCap-COCO	**32.60**	**27.50**	**108.55**	**20.33**
DiscriTune-COCO	32.31	26.05	105.40	20.03
w/ 5 hard distractors	29.85	25.53	100.25	19.50
w/ 10 hard distractors	29.20	25.25	98.69	19.30
Conceptual Captions				
ClipCap-COCO	1.47	6.43	23.74	7.98
DiscriTune-COCO	**1.71**	**6.58**	**28.01**	**9.00**
w/ 5 hard distractors	1.47	6.22	25.11	8.50
w/ 10 hard distractors	1.39	6.15	24.69	8.40
Flickr				
ClipCap-COCO	17.21	18.43	41.65	12.04
DiscriTune-COCO	18.48	18.61	44.78	12.68
w/ 5 hard distractors	**18.75**	**18.95**	**45.15**	**13.00**
w/ 10 hard distractors	18.23	18.68	44.28	12.85

Table 5. NLG metrics (BLEU@4, METEOR, CIDEr and SPICE) for ClipCap and ClipCap-based DiscriTune captions on COCO, ConceptualCaptions and Flickr, after training with 5 or 10 automatically mined hard distractors and testing with randomly selected ones.

Table 6 we report NLG metrics and retrieval accuracy, respectively. Overall, we see a mixed picture. With respect to the NLG metrics, hard distractors are helpful only for one of the two OOD datasets (Flickr), but at the price of a larger performance drop in-domain (COCO). Concerning retrieval accuracy, hard distractors give a slight improvement over random ones for in-domain data (COCO) and only on Flickr but not on Conceptual Captions for out-of-domain data.

We conjecture that the harder setup can lead to overfitting the quirks of the frozen retriever, in some cases leading to (slightly) poorer generalization. Studying the impact of the retrieval task with respect to number and type of distractors is an interesting direction for future work.

COCO	
Model	P@1
ClipCap-COCO	74.2
DiscriTune-COCO	84.8
w/ 5 hard distractors	84.9
w/ 10 hard distractors	**85.2**
Conceptual Captions	
ClipCap-COCO	73.0
DiscriTune-COCO	**83.6**
w/ 5 hard distractors	82.3
w/ 10 hard distractors	82.0
Flickr	
ClipCap-COCO	65.9
DiscriTune-COCO	79.4
w/ 5 hard distractors	**79.8**
w/ 10 hard distractors	79.1

Table 6. P@1 retrieval accuracy for ClipCap and ClipCap-based DiscriTune captions on COCO, Conceptual Captions and Flickr, after training with 5 or 10 automatically mined hard distractors and testing with randomly selected ones.

E. Caption Analysis: Nouns

The patterns we encountered in the adjective analysis presented in section 5 are confirmed by the noun lemma analysis. In Table 7, we report the top 10 noun lemmas most strongly associated with the human, Clipcap and DiscriTune captions in the Conceptual Captions dataset. DiscriTune favours words with strong and precise visual content, such as *dress*, *woman*, *pair*, *garden*, *field* and *lake*. Human captions tend to include more generic terms such as *person* and *background*, as well as several nouns that might be describing images at a more abstract level, that would probably not favour their precise identification (*image, time, summer, style, part*). The preference for these more abstract terms is even more pronounced in ClipCap (*view, actor, portrait, illustration, premiere, artist, vector, property*).

human	*ClipCap*	*DiscriTune*
	nouns	
person	view	forest
time	illustration	night
day	actor	dress
water	premiere	garden
image	person	celebrity
summer	artist	field
instrument	portrait	woman
style	vector	lake
background	background	pair
part	property	group

Table 7. Top 10 noun lemmas most associated to a caption type (*human*, *ClipCap* or *DiscriTune*) according to the local Mutual Information association statistics computed on all captions generated for our full Conceptual Captions test set.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 382–398, Cham, 2016. Springer International Publishing. 3

[2] Manuele Barraco, Matteo Stefanini, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. CaMEL: Mean Teacher Learning for Image Captioning. In *International Conference on Pattern Recognition*, 2022. 2

[3] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. 3

[4] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of EMNLP*, pages 7514–7528, Punta Cana, Dominican Republic, 2021. 2

[5] Benno Krojer, Vaibhav Adlakha, Vibhav Vineet, Yash Goyal, Edoardo Ponti, and Siva Reddy. Image retrieval from contextual descriptions. In *Proceedings of ACL*, pages 3426–3440, Dublin, Ireland, 2022. 3

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 3

[7] Jonathan Peirce, Jeremy R. Gray, Sol Simpson, Michael MacAskill, Richard Höchenberger, Hiroyuki Sogo, Erik Kastman, and Jonas Kristoffer Lindeløv. PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, 51(1):195–203, Feb. 2019. 1

[8] R. Vedantam, C. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. 2, 3

[9] Ronald Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 2