Supplementary: Integrating Language Guidance into Vision-based Deep Metric Learning

A. Experimental Details

For all our experiments, we utilize PyTorch [77]. Underlying backbones and training protocols are adapted from previous research (e.g. [66, 72, 82, 90, 91, 110]) including the codebase provided through [91]. More specifically, our experiments utilize either a ResNet50 [41] with embedding dimensionality of 128 or 512 as well as an Inception-BN [101] with dimensionality 512. The ImageNet-pretrained network weights were taken from `timm` [112] as well as `torchvision` [77].

Both for studies of relative improvements as well as state-of-the-art performance comparisons, optimization is done using Adam [53] with a base learning rate of 10^{-5}, consistent weight decay [57] of $3 \cdot 10^{-4}$ and batchsizes between 80 and 112. Our relative evaluation follows the protocol proposed in [91], while for our state-of-the-art comparison, we provide a thorough evaluation against different literature methods separated by the utilized underlying backbone. For our language backbone, we chose the language-part of CLIP [83] (specifically the "ViT-B/32" variant) and the provided tokenizer, but show in section 4.3 that essentially any big language model can be used for this task. This shows that improvements are not based on potential minor dataset overlap in the image-part of the CLIP training protocol and potential implicit information bleeding into the language model. We also note that the authors of [83] themselves highlight that even with data overlap, performance is not impacted in a relevant fashion. Applying language-guidance to S2SD [90], we found placing more emphasis on the feature distillation part instead of the dimensionality distillation worked better in conjunction with both *ELG* and *PLG*. To avoid large-scale hyperparameter grid searches, we thus simply set the weights for dimensionality matching to zero and only adjust the feature distillation weight.

For the scaling ω of our language-guidance (see Eq. 4), we found $\omega \in [1, 10]$ to work consistently for our experiments on CARS196 and CUB200-2011 and $\omega \in [0.1, 1]$ on Stanford Online Products, which accounts for the magnitude of the base loss function $\mathcal{L}_{\text{DML}}$. For the state-of-the-art study in §4.2, we found these parameter values to transfer well to the other backbones and embedding dimensionalities.

B. Additional Experimental Results

B.1. Additional language models

To study the impact of language guidance provided with different pretrained language models, Table S2 provides an extensive evaluation of different language model architectures and pretrainings. As can be seen, performance boosts are consistent, regardless of the exact choice of language model, supporting the general benefit of language as an auxiliary, performance-facilitating modality for finegrained visual similarity tasks.

B.2. Conceptual approaches to language inclusion

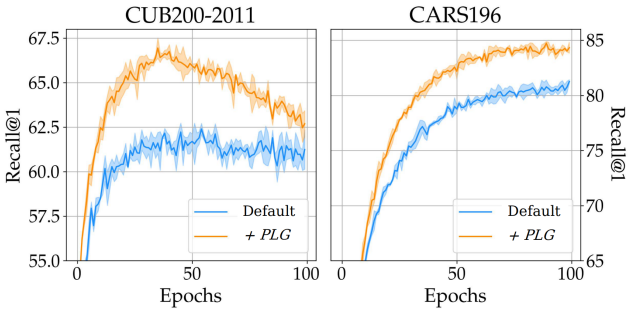

Figure S1. *Impact on convergence.* Matching performance is reached much earlier, with significantly better overall downstream generalization performance.

To directly incorporate language context into a discriminative DML objective, we utilize the language similarities to either adjust the mining mask or the loss scale in the multisimilarity loss [110]. To adjust the mining mask, given an anchor sample x_a, positives x_p and negatives x_n are selected if, respectively,

$$\begin{aligned} f\left(S^{\text{lang}}_{a,p}, s(\psi, \psi_p)\right) &< \min_{\psi_k \in \mathcal{B}, y_k \neq y_a} s(\psi_a, \psi_k) + \epsilon \\ f\left(S^{\text{lang}}_{a,n}, s(\psi_a, \psi_n)\right) &< \min_{\psi_k \in \mathcal{B}, y_k = y_a} s(\psi_a, \psi_k) - \epsilon \end{aligned} \tag{S1}$$

with similarity function $s(\bullet, \bullet)$ and language similarity scaling $f(\bullet, \bullet)$. For $f(\bullet, \bullet)$, we investigate different orders of interpolation

$$f(S^{\text{lang}}_{ij}, S^{\text{img}}_{ij})_{\nu_1, \nu_2} = \left[(1 - \nu_1) \cdot {S^{\text{lang}}_{ij}}^{\nu_2} + \nu_1 \cdot {S^{\text{img}}_{ij}}^{\nu_2}\right]^{1/\nu_2} \tag{S2}$$

to adjust between sole visual similarity and language similarity. To re-weight loss components (with positive and neg-

Table S1. *Relative comparison.* We follow protocols proposed in [91][6], with no learning rate scheduling, to ensure exact comparability. The results show significant improvements on CUB200 and CARS196 when language-guidance is applied (with expert- and pseudolabels). (*) For SOP, only 12 superlabels are given for 11,318 training classes. Similarly, SOP only contains very few samples per class, making pseudolabel class estimates very noisy. This makes the benefits of language guidance limited.

BENCHMARKS→	CUB200-2011			CARS196			SOP(*)		
APPROACHES ↓	R@1	NMI	mAP@1000	R@1	NMI	mAP@1000	R@1	NMI	mAP@1000
Multisimilarity	62.8 ± 0.2	67.8 ± 0.4	31.1 ± 0.3	81.6 ± 0.3	69.6 ± 0.5	31.7 ± 0.1	76.0 ± 0.1	89.4 ± 0.1	43.3 ± 0.1
+*ELG*	67.3 ± 0.2	69.6 ± 0.6	34.8 ± 0.2	85.3 ± 0.1	71.7 ± 0.2	32.7 ± 0.2	76.0 ± 0.2	89.5 ± 0.1	43.5 ± 0.1
+*PLG* Top-5	67.1 ± 0.4	69.6 ± 0.6	34.6 ± 0.5	85.4 ± 0.2	71.3 ± 0.1	32.8 ± 0.2	76.4 ± 0.1	89.6 ± 0.1	43.7 ± 0.1
Margin, $\beta = 1.2$	62.7 ± 0.6	68.0 ± 0.3	32.2 ± 0.3	79.4 ± 0.5	66.6 ± 0.7	32.8 ± 0.2	78.0 ± 0.3	90.3 ± 0.2	46.3 ± 0.2
+*ELG*	65.3 ± 0.5	68.5 ± 0.4	33.5 ± 0.3	83.2 ± 0.5	69.0 ± 0.6	33.4 ± 0.3	77.8 ± 0.1	90.2 ± 0.1	46.1 ± 0.1
+*PLG* Top-5	65.2 ± 0.5	68.5 ± 0.4	33.5 ± 0.3	83.4 ± 0.4	69.1 ± 0.4	33.7 ± 0.3	78.3 ± 0.2	90.3 ± 0.2	46.5 ± 0.1
Multisimilarity + S2SD	67.7 ± 0.3	71.5 ± 0.2	35.5 ± 0.2	86.5 ± 0.1	71.4 ± 0.4	35.1 ± 0.3	77.7 ± 0.2	89.9 ± 0.1	45.3 ± 0.3
+*ELG*	68.9 ± 0.4	72.5 ± 0.3	36.4 ± 0.5	88.2 ± 0.2	72.0 ± 0.1	36.0 ± 0.1	77.8 ± 0.1	90.0 ± 0.2	45.3 ± 0.2
+*PLG* Top-5	69.0 ± 0.4	72.4 ± 0.2	36.6 ± 0.3	88.4 ± 0.3	72.4 ± 0.2	36.2 ± 0.2	78.0 ± 0.1	90.0 ± 0.1	45.6 ± 0.1

Table S2. *Models vs. guidance quality.* Performance improves regardless of the exact large pretrained language model. Strong improvements can even be achieved through large-scale pretrained word embeddings such as FastText [5] and GloVe [79]. However, using less transferable word hierarchies falls short in comparison.

BENCHMARKS→	CUB200-2011		CARS196	
MODELS ↓	R@1	mAP @1000	R@1	mAP @1000
Baseline	62.8 ± 0.2	31.1 ± 0.3	81.6 ± 0.3	31.7 ± 0.1
+ CLIP-L [83]	67.3 ± 0.2	**34.8 ± 0.2**	85.3 ± 0.1	**32.7 ± 0.2**
(a) Language Models				
+ BERT [23]	66.9 ± 0.3	33.5 ± 0.2	84.9 ± 0.1	32.3 ± 0.1
+ DistBert [93]	66.7 ± 0.1	33.4 ± 0.2	85.4 ± 0.4	32.4 ± 0.1
+ Roberta-B [64]	67.0 ± 0.2	33.8 ± 0.2	84.9 ± 0.1	32.3 ± 0.3
+ Roberta-L [64]	67.3 ± 0.2	33.9 ± 0.3	85.1 ± 0.2	32.4 ± 0.2
+ DistRoberta [113]	66.0 ± 0.2	32.2 ± 0.2	85.0 ± 0.3	32.1 ± 0.2
+ Reformer [54]	66.7 ± 0.1	33.1 ± 0.1	85.5 ± 0.2	32.0 ± 0.2
+ MPNet [98]	66.2 ± 0.3	32.3 ± 0.2	85.4 ± 0.2	32.3 ± 0.3
+ GPT2 [84]	67.0 ± 0.3	33.7 ± 0.1	84.8 ± 0.4	32.4 ± 0.1
+ Top 3	**67.5 ± 0.2**	34.5 ± 0.3	**85.6 ± 0.3**	32.5 ± 0.3

ative pairs for anchor x_a, $\mathcal{P}_a^+$ and $\mathcal{P}_a^-$), we instead compute

$$\mathcal{L}_{\text{MSIM}}^{ELG} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\frac{1}{\alpha}\log\left(1+\sum_{k\in\mathcal{P}_i^+} e^{-\alpha\left(\frac{S_{ik}^{\text{lang}}}{S_{ik}^{\text{img}}}\right)^{\nu_3}(S_{ik}^{\text{img}}-\lambda)}\right) + \frac{1}{\beta}\log\left(1+\sum_{k\in\mathcal{P}_i^-} e^{\beta\left(\frac{S_{ik}^{\text{lang}}}{S_{ik}^{\text{img}}}\right)^{\nu_4}(S_{ik}^{\text{img}}-\lambda)}\right) \tag{S3}$$

thus providing a scaling to the utilized visual similarity S_{ik}^{img} based on the (relative) similarity to the respective language similarity. In all cases, a grid search both over newly introduced hyperparameters (ν_1, ν_2, ν_3 and ν_4) as well as the default multisimilarity loss parameters (α, β, λ) is performed.

For the language similarity scaling $f(\bullet,\bullet)$, we found linear interpolation ($\nu_1 = \nu_2 = 1$) to work best. For $\mathcal{L}_{\text{MSIM}}^{ELG}$, we found $\nu_3 = \nu_4 = 0.75$ to work well, but had to readjust $\alpha = 1.5$ and $\beta = 45$ slightly to account for the change in magnitude.

For our matching objective, in which we incorporate language context by training either a MLP over embeddings or a transformer (ViT, [24]) over a sequence of network features to predict language embeddings ψ_{lang}, the respective networks are trained following

$$\mathcal{L}_{\text{Match}}(\psi_i, \phi_i, \psi_{\text{lang},i}) = g_\rho^{\text{match}}(\psi_i, \phi_i)^T \psi_{\text{lang},i} \tag{S4}$$

with $g_\rho^{\text{match}}(\psi_i, \phi_i)$ denoting the unit-normalized mapping from embedding/feature space to (normalized) language space using either the MLP or ViT with parameters ρ.

Finally, as a base reference, we also investigate CLIP-style training in which we utilize direct contrastive training between image and language embeddings following [83] as regularizer against $\mathcal{L}_{\text{DML}}$:

$$\mathcal{L}_{\text{CLIP}}(S^{\text{mixed}}) = \frac{1}{|\mathcal{B}|}\sum_{i}^{|\mathcal{B}|} -\frac{1}{2}\log\left(\sigma\left(S_{i,:}^{\text{mixed}}\cdot e^T\right)_i\right) - \frac{1}{2}\log\left(\sigma\left(S_{:,i}^{\text{mixed}}\cdot e^T\right)_i\right) \tag{S5}$$

with similarity matrix between minibatch image and language embeddings S^{mixed}.

B.3. Language guidance from pseudolabels

For *PLG*, we investigate performance dependence on the number of top-k pseudolabels assigned to every class and their inclusion into training (§3.3. Table S3 highlights that more pseudolabels benefit generalization (optimum for $k \in [5, 10]$), that distillation from a single averaged similarity matrix (see Eq. 5) performs better than (or comparable

Table S3. *Ablations on Pseudolabel guidance.* More pseudolabels per class improve generalization performance, with class-level pseudolabelling and distillation from a single average language similarity matrix offering highest improvements.

BENCHMARKS→	CUB200-2011		CARS196	
APPROACHES ↓	R@1	mAP @1000	R@1	mAP @1000
Baseline	62.8 ± 0.2	31.1 ± 0.2	81.6 ± 0.3	31.7 ± 0.1
+ *ELG*	67.3 ± 0.2	34.8 ± 0.2	85.3 ± 0.1	32.7 ± 0.2
Number of pseudolabels				
+ *PLG*	66.2 ± 0.6	33.8 ± 0.2	85.2 ± 0.1	32.7 ± 0.1
+ *PLG* (Top-5)	67.1 ± 0.4	34.6 ± 0.5	85.4 ± 0.2	32.8 ± 0.2
+ *PLG* (Top-10)	67.2 ± 0.2	34.5 ± 0.3	85.3 ± 0.2	32.4 ± 0.3
+ *PLG* (Top-20)	67.0 ± 0.1	34.5 ± 0.2	84.9 ± 0.4	32.1 ± 0.3
Word Hierarchies				
PLG+WordNet (Top-5)	64.4 ± 0.1	31.5 ± 0.2	82.9 ± 0.2	31.3 ± 0.2
pseudo-HierMatch	65.0 ± 0.2	32.3 ± 0.2	82.8 ± 0.3	31.6 ± 0.2
Different pseudolabel matching methods				
Sample (Top-5)	67.0 ± 0.1	34.2 ± 0.1	85.2 ± 0.2	32.2 ± 0.3
Dense (Top-5)	67.0 ± 0.2	33.7 ± 0.4	84.0 ± 0.1	31.5 ± 0.4
Multi (Top-5)	67.2 ± 0.1	34.3 ± 0.2	85.1 ± 0.2	32.2 ± 0.1
Dense + Multi	66.0 ± 0.3	34.1 ± 0.2	84.3 ± 0.3	31.9 ± 0.2

to) joint distillation from each pseudolabel similarity matrix ("*Multi (Top-5)*"), and that is does not matter if pseudolabels are computed for classes or individual samples ("*Sample*").

In addition, we study whether computing a pseudolabel similarity matrix for each pseudolabel pairing, disregarding the ordering[7], benefits overall performance ("*Dense (Top-5)*" and "*Dense + Multi*"), but found no notable benefit. Furthermore, Table S3 shows that leveraging hierarchies as described in §4.3 also performs notable worse in the pseudolabel domain. Finally, we find impact on overall training time of *PLG* to be negligible, while convergence are in parts even improved (see Supp.-Fig. S1).

B.4. Convergence of *PLG* models

Figure S1 shows that *PLG (Top-5)* allows underlying objectives to reach similar performance after significantly less training, with much higher overall performance after full training. With *PLG* only requiring an initial forward pass of training samples through the ImageNet-pretrained backbone and of all unique classnames through the language model, impact on overall training time is also negligible.

[7]E.g. for $k = 5$, "*Dense*" introduces $k^2 = 25$ target matrices.