This supplementary material consists of five sections. In Section A, we provide the pseudocode for Sieve. In Section B, we present the per-task performance gain of Sieve and Sieve+CLIPScore relative to CLIPScore on *large* scale. In Section C, we present Sieve's capability to augment other data pruning methods: Section C.1, investigating the effect of filtering via text-spotting on Sieve, and Section C.2 investigating the effect of distribution alignment with ImageNet on Sieve. Finally, in Section D, we show the superiority of our sentence transformer for textual semantic clustering compared to CLIP and BLIP text encoders.

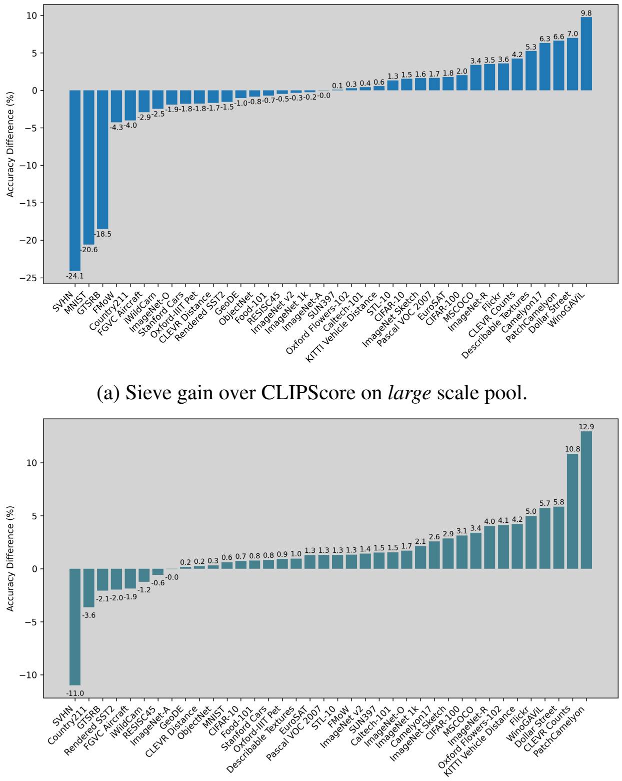
A. Pseudocode

Algorithm 1 Sieve Pseudocode to filter image-text datasets.

```
Require: Dataset $\mathcal{D}$, Fraction $k$ fraction to prune, Caption generator $G$, Sentence transformer $S$, Set of medium phrases $\mathcal{M}$, Number of captions to generate $r$
Ensure: Pruned dataset $\mathcal{D}_{\text{Sieve}}$
for each $I, T$ in $\mathcal{D}$ do    ▷ Loop over image-text pairs in the dataset
    $e \leftarrow S(T)$    ▷ Obtain sentence embedding of text label
    $\mathcal{V} \leftarrow \{\}$    ▷ Initialize empty set of scores
    $\mathcal{T}^G \leftarrow G(I, r)$    ▷ Generate a set, $\mathcal{T}^G$, of $r$ captions
    for each $T^G$ in $\mathcal{T}^G$ do    ▷ Loop over generated captions
        $T^{G'} \leftarrow M(T^G)$    ▷ Mask medium words
        $e^G \leftarrow S(T^{G'})$    ▷ Obtain sentence embeddings of generated caption
        $v \leftarrow \langle e^G, e \rangle$    ▷ Compute sentence similarity
        $\mathcal{V} += v$    ▷ Append to set of scores
    end for
    $f_{\text{Sieve}}(I, T) \leftarrow \max(\mathcal{V})$    ▷ Obtain maximum score
end for
Rank $\mathcal{D}$ by $f_{\text{Sieve}}(I, T)$ in descending order to obtain $\texttt{rank}_{\text{Sieve}}(x)$
$\mathcal{D}_{\text{Sieve}} \leftarrow$ top $k\%$ of $\mathcal{D}$ based on $\texttt{rank}_{\text{Sieve}}(x)$
return $\mathcal{D}_{\text{Sieve}}$
```

B. Per-task Performance (Large)

Figure 6 shows the change in accuracy introduced by Sieve as well as Sieve+CLIPScore on each task compared to CLIPScore on *large* scale. We observe similar patterns of relative gain for Sieve as observed on the *medium* scale, where Sieve achieves superior performance on retrieval and on medical diagnosis tasks relative to CLIPScore. In addition, when fused with CLIPScore, retrieval performance is boosted while the limited performance of Sieve on tasks that require OCR like SVHN [25] and MNIST [19] is significantly improved. Finally, Sieve with and without fusion with CLIPScore achieves a large boost on DollarStreet [35], a dataset that shows pictures of household items from families of diverse ethnic and economic backgrounds. This boost demonstrates that CLIP models pretrained with Sieve filtered image-text datasets is better at interpreting a wide array of real-world scenes and objects.



(a) Sieve gain over CLIPScore on *large* scale pool.

(b) Sieve+CLIPScore gain over CLIPScore on *large* scale pool.

Figure 6. The relative performance gain of Sieve and Sieve+CLIPScore relative to CLIPScore on 38 downstream tasks on "large" scale pool.

C. Combining Sieve with Other Pruning Methods

C.1. Sieve ∩ Text Spotting

First, we study the effect of Sieve on Text-spotting. Text-spotting involves detecting and recognizing text in images and filtering image-text pairs with high overlap between spotted text (text detected in image) and alt-text (associated label of image) [30]. T-MARS [24] uses text-spotting to reduce the CLIPScore of image-text pairs with high intersection between text in images and their corresponding text. For text spotting, we first utilize a text detector, to detect and compute the percentage of image covered by text. Next, we rank samples in ascending order of the percentage of pixels covered by text. Finally, we keep the top 80% of the sam-

Filter	ImageNet	ImageNet dist. shift	VTAB	Ret.	Avg.
CLIPScore	27.30	23.00	33.80	25.10	32.80
Sieve	**29.60**	**24.93**	**35.07**	**28.57**	**34.03**
CLIPScore ∩ Text Spotting	29.75	24.10	**35.65**	24.95	**34.05**
Sieve ∩ Text Spotting	**30.10**	**25.05**	34.15	**28.35**	33.90

Table 6. Intersection of samples ranked by Sieve and samples kept by Text spotting filter on *medium* scale.

Filter	ImageNet	ImageNet dist. shift	VTAB	Ret.	Avg.
CLIPScore	57.8	47.4	53.8	46.6	52.9
Sieve	57.3	47.8	52.0	52.0	52.3
Sieve+CLIPScore	59.7	49.1	**54.8**	51.1	**54.6**
CLIPScore ∩ ImageNet	63.1	50.8	54.6	49.8	53.7
Sieve ∩ ImageNet	61.2	49.2	51.3	51.4	51.4
Sieve+CLIPScore ∩ ImageNet	**63.8**	**51.4**	53.1	**53.3**	53.6

Table 7. Intersection of ImageNet-based filtering on *large* scale. We achieve state-of-the-art performance on ImageNet, ImageNet out-of-distribution and retrieval tasks.

Top-k%	Sieve	Sieve + CLIPScore
10	16.00%	27.18%
20	29.99%	44.00%
30	39.54%	56.93%

Table 8. Intersection-Over-Union between the unique ids of top-k samples selected by CLIPScore and 1) Sieve, 2) Sieve+CLIPScore on *medium* scale

ples and intersect either with top 30% CLIPScore or top 20% top Sieve samples. In Table 6, we observe that CLIPScore filtering gains from Text-spotting more than Sieve. In addition, the performance of CLIPScore ∩ Text-spotting is close to Sieve on its own. It is important to note that we start with top 20% of the samples (24M) for Sieve, we end up with fewer samples for pretraining with Sieve (19.2M) compared to top 30% from CLIPScore (24M). For future work, we are planning to implement text-spotting tightly coupled with Sieve by masking detected text in images prior to the captioning step.

C.2. Sieve ∩ ImageNet

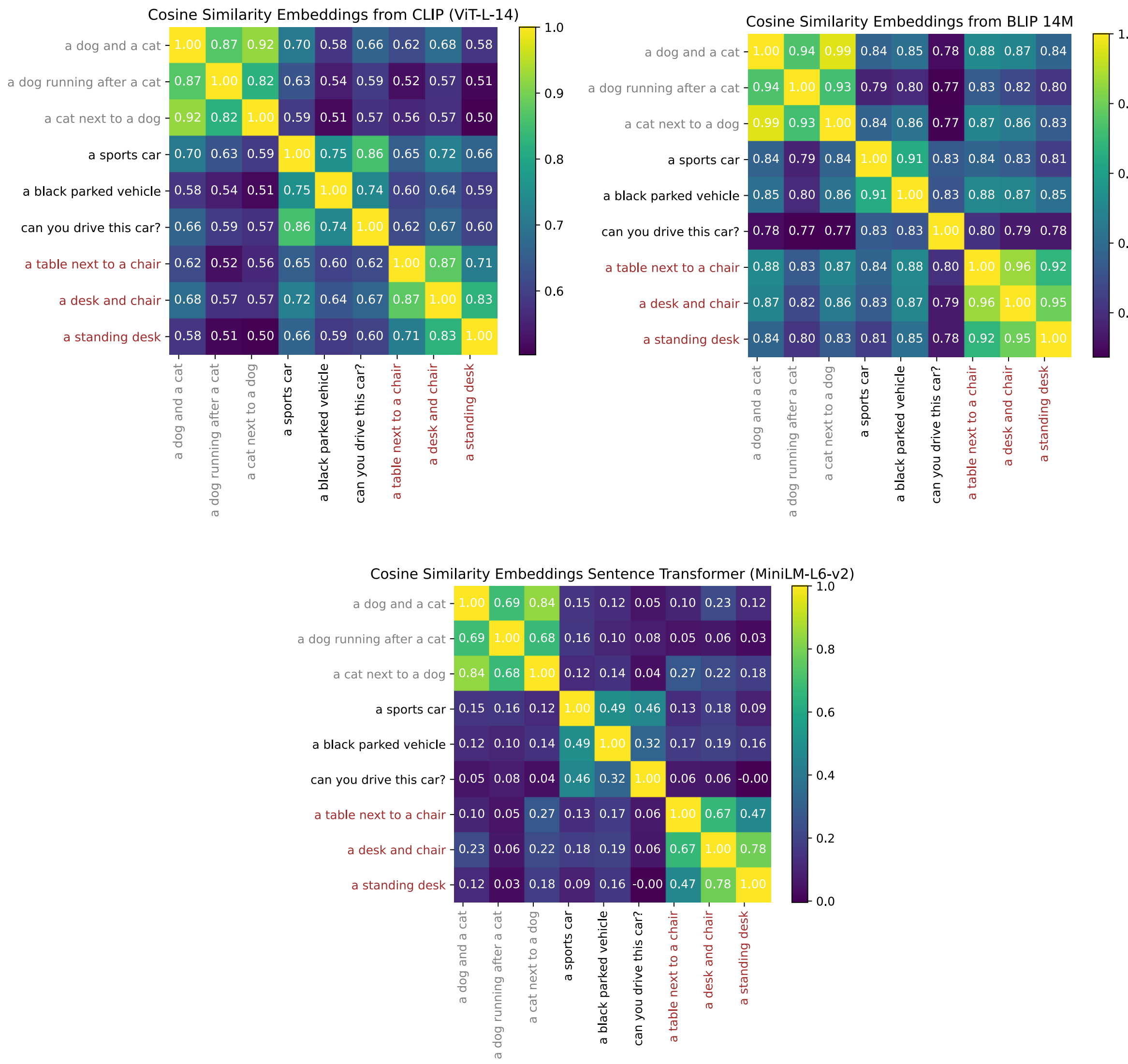
One of the common approaches to pruning large-scale noisy image-text datasets is by sampling image-text pairs where the image is clustered near pretrained CLIP embeddings of ImageNet training set samples [9]. This distribution alignment can result in improved visual representations at the expense of generalization. We generate a pretraining dataset using Sieve and then investigate the utility of only keeping Sieve samples intersecting with ImageNet samples or Sieve+CLIPScore (fused) samples intersecting with ImageNet samples. Looking at Table 7, compared to Sieve, we observe that the intersection of Sieve and ImageNet leads to significant improvements on ImageNet but at the expense of average performance. We also observe that compared to CLIPScore ∩ ImageNet, Sieve+CLIPScore ∩ ImageNet leads to the best performance on ImageNet, ImageNet out-of-distribution shift and a significant improvement on retrieval tasks (+3.5%).

D. Textual Semantic Clustering

We investigate the textual semantic clustering abilities of three models; Sentence Transformer, CLIP text encoder and BLIP text encoder. We create 3 groups of sentences, where each group contains 3 sentences describing similar objects. The goal is to study which text encoder is best at distinguishing between sentences from the same group. The best embeddings space for estimating the alignment between generated caption and alt-text, should have a representation space, where semantically similar sentences are close to each other, while semantically-distinct sentences are far away. In Figure 7, we depict the cosine similarity values between the embeddings of all sentences from the 3 groups, where sentences from the same group have the same color. We observe that compared to CLIP and BLIP text embedding space, Sentence transformer has the best intra and inter-cluster values and therefore results in the best alignment score.

Scale	Filtering	Dataset Size	ImageNet	ImageNet dist. shifts	VTAB	Retrieval	Average over 38 datasets
Small (12.8 Million)	No Filtering	12.8M	2.5	3.3	14.5	11.4	13.2
	Basic Filtering	3.0M	3.8	4.3	15.0	11.8	14.2
	LAION Filtering	1.3M	3.1	4.0	13.6	9.2	13.3
	CLIPScore	3.8M	5.1	5.5	19.0	11.9	17.3
	Sieve	3.0M	6.0	6.2	18.2	13.6	17.4
	Sieve+CLIPScore	2.4M	6.0	6.0	19.8	13.3	**18.0**

Table 9. Zero-shot performance of CLIP models pretrained using filtering strategies on *small* scale pools of the DataComp benchmark.



Cosine Similarity Embeddings from CLIP (ViT-L-14)

	a dog and a cat	a dog running after a cat	a cat next to a dog	a sports car	a black parked vehicle	can you drive this car?	a table next to a chair	a desk and chair	a standing desk
a dog and a cat	1.00	0.87	0.92	0.70	0.58	0.66	0.62	0.68	0.58
a dog running after a cat	0.87	1.00	0.82	0.63	0.54	0.59	0.52	0.57	0.51
a cat next to a dog	0.92	0.82	1.00	0.59	0.51	0.57	0.56	0.57	0.50
a sports car	0.70	0.63	0.59	1.00	0.75	0.86	0.65	0.72	0.66
a black parked vehicle	0.58	0.54	0.51	0.75	1.00	0.74	0.60	0.64	0.59
can you drive this car?	0.66	0.59	0.57	0.86	0.74	1.00	0.62	0.67	0.60
a table next to a chair	0.62	0.52	0.56	0.65	0.60	0.62	1.00	0.87	0.71
a desk and chair	0.68	0.57	0.57	0.72	0.64	0.67	0.87	1.00	0.83
a standing desk	0.58	0.51	0.50	0.66	0.59	0.60	0.71	0.83	1.00

Cosine Similarity Embeddings from BLIP 14M

	a dog and a cat	a dog running after a cat	a cat next to a dog	a sports car	a black parked vehicle	can you drive this car?	a table next to a chair	a desk and chair	a standing desk
a dog and a cat	1.00	0.94	0.99	0.84	0.85	0.78	0.88	0.87	0.84
a dog running after a cat	0.94	1.00	0.93	0.79	0.80	0.77	0.83	0.82	0.80
a cat next to a dog	0.99	0.93	1.00	0.84	0.86	0.77	0.87	0.86	0.83
a sports car	0.84	0.79	0.84	1.00	0.91	0.83	0.84	0.83	0.81
a black parked vehicle	0.85	0.80	0.86	0.91	1.00	0.83	0.88	0.87	0.85
can you drive this car?	0.78	0.77	0.77	0.83	0.83	1.00	0.80	0.79	0.78
a table next to a chair	0.88	0.83	0.87	0.84	0.88	0.80	1.00	0.96	0.92
a desk and chair	0.87	0.82	0.86	0.83	0.87	0.79	0.96	1.00	0.95
a standing desk	0.84	0.80	0.83	0.81	0.85	0.78	0.92	0.95	1.00

Cosine Similarity Embeddings Sentence Transformer (MiniLM-L6-v2)

	a dog and a cat	a dog running after a cat	a cat next to a dog	a sports car	a black parked vehicle	can you drive this car?	a table next to a chair	a desk and chair	a standing desk
a dog and a cat	1.00	0.69	0.84	0.15	0.12	0.05	0.10	0.23	0.12
a dog running after a cat	0.69	1.00	0.68	0.16	0.10	0.08	0.05	0.06	0.03
a cat next to a dog	0.84	0.68	1.00	0.12	0.14	0.04	0.27	0.22	0.18
a sports car	0.15	0.16	0.12	1.00	0.49	0.46	0.13	0.18	0.09
a black parked vehicle	0.12	0.10	0.14	0.49	1.00	0.32	0.17	0.19	0.16
can you drive this car?	0.05	0.08	0.04	0.46	0.32	1.00	0.06	0.06	-0.00
a table next to a chair	0.10	0.05	0.27	0.13	0.17	0.06	1.00	0.67	0.47
a desk and chair	0.23	0.06	0.22	0.18	0.19	0.06	0.67	1.00	0.78
a standing desk	0.12	0.03	0.18	0.09	0.16	-0.00	0.47	0.78	1.00

Figure 7. Confusion matrix of cosine similarity illustrating the performance of each text encoder in similarity. We show 9 sentences split into 3 groups of consecutive sentences, where sentences within each group describe similar concepts. We observe that Sentence Transformer has better semantic textual clustering compared to CLIP and BLIP text encoders.