A. Evaluation Protocol Details

In the main text in Sec. 4.1, we described the motivation and methodology for choosing our *StableEval* set of 27 evaluations. We also categorized our main results in Tab. 2 into IN-shift, Object-Centric, and Scene-Centric, under the zero-shot classification section. We now provide additional details for these sections.

***StableEval* Protocol.** For rigorously defining our final evaluation suite, we first selected 34 candidate evaluation datasets popularly used for evaluating standard image-text contrastive pretraining [48, 84, 119] and adaptation [49, 129, 152, 193, 198] methods. These datasets ranged from standard natural image-classification, to fine-grained classification of birds, animals, and cars etc., to different domains of images like satellite imagery and street signs. The full set of 34 candidate evaluations we started with are: FGVC-Aircrafts [104], Oxford Flowers-102 [112], Oxford-IIIT Pets [115], Stanford Cars [78], Food-101 [15], Caltech-101 [86], CIFAR-10 [79], CIFAR-100 [79], Pascal VOC 2007 [36], EuroSAT [60], RESISC45 [64], STL-10 [28], SUN-397 [174], Dollar Street [127], GeoDE [121], Country211 [119], FMoW [26], DTD [27], iWildCam [10], PatchCamelyon [157], CLEVR Counts [72], CLEVR Distance [72], KITTI Distance [51], ImageNet-V2 [125], ImageNet-A [62], ImageNet-R [61], ObjectNet [8], ImageNet-Val [30], ImageNet-Sketch [164], Rendered SST2 [119], Flickr30k (I2T and T2I) [116], MSCOCO ((I2T and T2I)) [96].

We then trained several variants of standard SigLIP and CLIP models with a ViT-S/32 image-encoder and a BERT-small text-encoder, to quantify the amount of variance present for each evaluation dataset, solely due to the random seed (*i.e.*, different initialization of model weights). Specifically, we first trained 5 IID-SigLIP models on both DataComp-1B and WebLI-1B for 3B examples seen (*i.e.*, with randomly sampling batches of data at each step) by only changing the random seed. Note that we ensured that the exact samples seen per step in the training process was fixed—that is, the only randomness across the 5 different seed runs was the model initialization. We also trained an IID-CLIP model for 5 seeds to add variation on the training objective to the set of models. We then get the average standard deviation of each evaluation dataset by first averaging over the 5 different random seeds per method (*i.e.*, DataComp-IID-SigLIP, DataComp-IID-CLIP, WebLI-IID-SigLIP), and then averaging over the 3 different combinations of methods. This average standard deviation is taken to be the variability of each evaluation, which is shown in Fig. 3. We also tested this variability across other settings by changing the patch-size of the image-encoder (from S/32 to S/16) and increasing the model size (from S/32 to B/32), and found the variability (standard deviation) per evaluation dataset to be consistent.

Equipped with these standard deviations per evaluation dataset, we then aim to prune out the set of highly unstable evaluations from the full set of 34 evaluations by taking inspiration from the continuous inverse-variance weighting (IVW) method [58]. We start with the lowest-variance evaluation (Country211 with 0.15% standard deviation), and progressively add evaluations in increasing order of their computed standard deviations, each time computing the variability of the average over the current set of evaluations. For a set of N evaluations, the variability of the average is computed as $std(E_1...E_N)=\sqrt{\frac{1}{N^2}\sum_i var(E_i)}$. At each step, we compare the variability of the average with the variability of the most reliable evaluation (*i.e.*, Country211 with 0.15% standard deviation), and prune out all evaluations beyond the critical point where the variability of the average becomes larger than the Country211 variability. This leaves us with a set of 27 evaluations that are both diverse as well as stable across different random seeds. The 7 evaluation datasets that were pruned out of the final set are: EuroSAT, CLEVR Counts, GTSRB, iWildCam, SVHN, KITTI Distance, CLEVR Distance, PatchCamelyon, and Rendered-SST2.

Categorization of Datasets. Having identified our stable set of evaluation datasets, we next categorize them into different brackets for easier parsing of the different capabilities of the models in Tab. 2. In Tab. 4, we showcase the breakdown of the different categories represented in Tab. 2 for all 27 evaluations. We categorize them into *object-centric* datasets like FGVC-Aircrafts or Stanford Cars, *scene-centric* datasets like SUN-397 or RESISC45, *Imagenet-based natural distribution shifts* like ImageNet-V2 or ObjectNet, and other *miscellaneous* evaluations like DTD or Country211. Finally, we also evaluate our models on image-text retrieval datasets like COCO and Flickr, both using text-to-image retrieval and image-to-text retrieval, as separate evaluation metrics.

Table 4. **Final *StableEval* Set of 27 evaluations.**

Category	Dataset	Task	Test set size	Number of classes
Object-Centric	FGVC-Aircrafts [104]	Aircraft recognition	3,333	100
	Oxford Flowers-102 [112]	Flower recognition	6,149	102
	Oxford-IIIT Pets [115]	Pet classification	3,669	37
	Stanford Cars [78]	Vehicle recognition	8,041	196
	Food-101 [15]	Food recognition	25,250	101
	Caltech-101 [86]	Object recognition	6,085	102
	CIFAR-10 [79]	Visual recognition	10,000	10
	CIFAR-100 [79]	Visual recognition	10,000	100
	Pascal VOC 2007 [36]	Object recognition	14,976	20
	STL-10 [28]	Visual recognition	8,000	10
Scene-Centric	SUN-397 [174]	Scene recognition	108,754	397
	GeoDE [121]	Object/scene recognition	12,488	40
	RESISC45 [64]	Satellite imagery recognition	6,300	45
	FMoW [26]	Satellite imagery recognition	22,108	62
Distribution-shifts	ImageNet-V2 [125]	Visual recognition	10,000	1,000
	ImageNet-A [62]	Visual recognition	7,500	200
	ImageNet-R [61]	Visual recognition	30,000	200
	ObjectNet [8]	Visual recognition	18,574	113
Misc.	ImageNet-Val [30]	Visual recognition	50,000	1,000
	ImageNet-Sketch [164]	Visual recognition	50,889	1,000
	DTD [27]	Texture classification	1,880	47
	DollarStreet [127]	Object recognition	3,503	58
	Country211 [119]	Geolocation	21,100	211
Retrieval	Flickr30k (I2T, T2I) [116]	Image and text retrieval	31,014	N/A
	MSCOCO (I2T, T2I) [96]	Image and text retrieval	5,000	N/A

B. Image-text contrastive Objectives

Here, we expand the full image-text pretraining objectives described in Sec. 3.1. The per-sample softmax image-text objective is primarily used for training CLIP [119] models, while the per-sample sigmoid objective is primarily used in training SigLIP [190] models:

$$\mathcal{L}_{\text{softmax}}(x_i; \mathcal{B}) = -\frac{1}{2}\left(\log p_{ii}^{\text{img}\to\text{txt}} + \log p_{ii}^{\text{txt}\to\text{img}}\right) \tag{6}$$

$$\mathcal{L}_{\text{sigmoid}}(x_i; \mathcal{B}) = -\left(\log p_{ii}^{\text{sig}} + \sum_{j=1, j\neq i}^{b} \log(1 - p_{ij}^{\text{sig}})\right) \tag{7}$$

C. Proofs for Active Curation as Implicit Distillation

In this section, we provide derivations for our theoretical results in Sec. 3.2 showcasing the equivalence between active data curation and knowledge distillation. We first show the proof for the case where we use easy-reference scoring for data-curation, followed by the learnability-scoring case, and finally showcase a generalized version of the proof.

Setup. Recollect from the main paper text in Sec. 3.2, that we are given an image-text pretraining dataset $\mathcal{D}$. The simple training approach is to sample uniformly random batches of data $\mathcal{B}$ (of size b), from $\mathcal{D}$ at each step t, and minimize $\mathcal{L} \in \{\mathcal{L}_{\text{softmax}}, \mathcal{L}_{\text{sigmoid}}\}$ (see Appendix B for full equations for the loss objectives). We call this baseline, minimizing $\hat{\mathcal{L}} = \frac{1}{b}\sum_{x_i \sim \mathcal{U}[\mathcal{D}]} \mathcal{L}(x_i; \mathcal{B})$ as the *IID-baseline* (θ_{IID}). Further, remember that in the *active data curation* setup, we employ a smarter way to select batches, using a pretrained *reference* model θ_{ref}. At each step t, we select a sub-batch $\mathcal{B}$ (size b) from a much larger super-batch $\mathcal{S}$ (size B) according to an *active selection distribution* $\mathcal{A}[\mathcal{S}]$.

Active Data Curation as Implicit Distillation (ACID). We now show formally that active curation can be cast as "implicit distillation" and should benefit from larger reference models. The model now minimizes $\hat{\mathcal{L}} = \frac{1}{b}\sum_{x_i \sim \mathcal{A}[\mathcal{S}]} \mathcal{L}(x_i; \mathcal{B})$, which in expectation is $\mathcal{E} = \mathbb{E}[\hat{\mathcal{L}}] = \sum_{x \in \mathcal{D}} a(x)\mathcal{L}(x; \mathcal{B})$ given that super-batches $\mathcal{S}$ are sampled uniformly. Recall that $\mathcal{L}(x; \mathcal{B}) = -\sum_{i=1}^{b} y_i(x) \log q_i(x)$, where y_i are the labels of the contrastive task and q_i are the probabilities induced by the pairwise similarities of the student θ. Let p_i be the probabilities induced by the reference model θ_{ref}. In the case of *easy-reference scoring* and the softmax loss, $a(x) = \frac{1}{Z}\exp\sum_{i=1}^{b} y_i(x) \log p_i(x) = \frac{1}{Z}p_{i^*}(x)$ where i^* is the index of the one-hot label $y(x)$. As such,

$$
\begin{aligned}
\mathcal{E}_{\text{easy-ref}} &= -\sum_{x \in \mathcal{D}} a(x) \sum_{i=1}^{b} y_i(x) \log q_i(x) \\
&= -\frac{1}{Z}\sum_{x \in \mathcal{D}} p_{i^*}(x) \sum_{i=1}^{b} y_i(x) \log q_i(x) \\
&= -\frac{1}{Z}\sum_{x \in \mathcal{D}} \sum_{i=1}^{b} p_{i^*}(x) y_i(x) \log q_i(x) \\
&= \frac{1}{Z}\sum_{x \in \mathcal{D}} \text{KD}[p(x) \cdot y(x); q(x)]
\end{aligned}
\tag{8}
$$

This demonstrates that by curating data according to the reference model θ_{ref}, we implicitly distill its knowledge via a novel data-driven objective, using a combination of model predictions and real labels as targets. We next prove the equivalence of data curation and knowledge-distillation, when using learnability-based scoring for our active data curation.

Learnability-based Data Curation is Hard Distillation. When using learnability-based prioritization, the active selection distribution $\mathcal{A}$ factorizes as $a^{\text{learn}} = \frac{1}{Z}\exp(s^{\text{learn}}) = \frac{1}{Z}\exp[\mathcal{L}(\cdot|\theta) - \mathcal{L}(\cdot|\theta_{\text{ref}})] = a^{\text{easy-ref}} \cdot a^{\text{hard-learn}}$ where $a^{\text{hard-learn}} = \frac{1}{Z}\exp[\mathcal{L}(\cdot|\theta)]$ prioritizes examples with high loss according to the student. Since easy-reference prioritization yields implicit distillation (*I-ACID*, Eq. (4)), learnability prioritization yields:

$$
\begin{aligned}
\mathcal{E}_{\text{learn}} &= \sum_{x \in \mathcal{D}} a^{\text{hard-learn}}(x) \cdot a^{\text{easy-ref}}(x)\mathcal{L}(x; \mathcal{B}) \\
&= \frac{1}{Z}\sum_{x \in \mathcal{D}} a^{\text{hard-learn}}(x)\text{KD}[p(x) \cdot y(x); q(x)]
\end{aligned}
\tag{9}
$$

This demonstrates that learnability-based active curation is equivalent to implicit distillation on hard examples ("H-ACID") according to the student model.

ACID for general learning objectives. In the general case (including sigmoid-contrastive learning, and combined image-to-text and text-to-image softmax contrastive learning), $y(x)$ contains a set of labels $y_i(x)$ such that $\sum_{i=1}^{b} y_i(x) = 1$. In this case

$a(x) = \frac{1}{Z} \exp \sum_{i=1}^{b} y_i(x) \log p_i(x) \leq \frac{1}{Z} \sum_{i=1}^{b} y_i(x) p_i(x) = \frac{1}{Z} \hat{p}(x)$ due to the convexity of the exponential. In particular,

$$\mathcal{E}_{\text{easy-ref}} = -\sum_{x \in \mathcal{D}} a(x) \sum_{i=1}^{b} y_i(x) \log q_i(x) \geq -\frac{1}{Z} \sum_{x \in \mathcal{D}} \hat{p}(x) \sum_{i=1}^{b} y_i(x) \log q_i(x) \tag{10}$$

$$\geq \frac{1}{Z} \sum_{x \in \mathcal{D}} \text{KD}[\hat{p}(x) \cdot y(x); q(x)] \tag{11}$$

As such, learning from actively-curated data minimizes an upper bound on the KD objective described previously, for general learning objectives of the form $\sum_{i=1}^{b} y_i(x) \log q_i(x)$, including the softmax- and sigmoid-contastive objectives we utilize in this work.

D. Knowledge Distillation Objectives

In this section, we describe in detail all the knowledge-distillation methods we use to compare as baselines in our results in Sec. 4.2. Given the student model θ and a pretrained teacher model θ_{teacher}, we considered three main objectives for distilling the knowledge from the teacher θ_{teacher} into the student model θ.

Softmax contrastive distillation. Here, our aim is to distill the contrastive logit matrix from the teacher to the student. Formally, given a data-batch B, we extract teacher embeddings $\{(z^{\text{img}}_{i,t}, z^{\text{txt}}_{i,t})\}$ and student embeddings $\{(z^{\text{img}}_{i,s}, z^{\text{txt}}_{i,s})\}$. The teacher and student contrastive matrices, $\mathcal{T}_{b\times b}$ and $\mathcal{S}_{b\times b}$, contain the teacher and student image-text logits, respectively:

$$\mathcal{T}_{i,j} = \alpha_t z^{\text{img}}_{i,t} \cdot z^{\text{txt}}_{j,t}, \mathcal{S}_{i,j} = \alpha_s z^{\text{img}}_{i,s} \cdot z^{\text{txt}}_{j,s} \tag{12}$$

Our softmax distillation objective takes the form of a cross-entropy loss between the teacher and student contrastive matrices, considering the texts as labels by applying a row-wise softmax on the contrastive matrices ($\mathcal{T}$, $\mathcal{S}$) and the images as labels by applying a column-wise softmax ($\mathcal{T}^T$, $\mathcal{S}^T$).

$$\mathcal{L}_{\text{smax-dist}} = -\frac{1}{2b}\sum_{i=1}^{b}\left(\underbrace{\texttt{softmax}(\mathcal{T}_{i,\cdot})\log\texttt{softmax}(\mathcal{S}_{i,\cdot})}_{\text{image-to-text}} + \underbrace{\texttt{softmax}(\mathcal{T}^T_{i,\cdot})\log\texttt{softmax}(\mathcal{S}^T_{i,\cdot})}_{\text{text-to-image}}\right) \tag{13}$$

Sigmoid contrastive distillation. Similarly as above, here we distill the teacher contrastive matrix into the student matrix. However, differently from the softmax case, in this loss we use the full teacher and student image-text logits with the addition of the bias term:

$$\mathcal{T}_{i,j} = \alpha_t z^{\text{img}}_{i,t} \cdot z^{\text{txt}}_{j,t} + \beta_t, \mathcal{S}_{i,j} = \alpha_s z^{\text{img}}_{i,s} \cdot z^{\text{txt}}_{j,s} + \beta_s \tag{14}$$

Our sigmoid distillation objective then simply takes the form a binary cross-entropy objective between the teacher and the student logits (converted to probabilites using the sigmoid (σ) activation):

$$\mathcal{L}_{\text{sig-dist}} = -\frac{1}{b}\sum_{i=1}^{b}\left(\sigma(\mathcal{T}_{i,\cdot})\log\sigma(\mathcal{S}_{i,\cdot}) + \sigma(-\mathcal{T}_{i,\cdot})\log\sigma(-\mathcal{S}_{i,\cdot})\right) \tag{15}$$

Feature-matching distillation. We also explore a distillation loss that directly aligns the image and text embeddings of the student and teacher models directly, using a simple mean-squared error. Such a strategy has also been explored in prior SoTA CLIP distillation works [180], with great efficacy. If the student and teacher embedding dimensions are different, we project the student embedding to the teacher dimension using a learnable linear projection head P_{head}:

$$\begin{aligned} \hat{z}^{\text{img}}_{i,s} = z^{\text{img}}_{i,s} P_{\text{head}}, \hat{z}^{\text{txt}}_{i,s} = z^{\text{txt}}_{i,s} P_{\text{head}} \\ \mathcal{L}_{\text{fm-dist}} = \frac{1}{2b}\sum_{i=1}^{b}\left(\underbrace{\|\hat{z}^{\text{img}}_{i,s} - z^{\text{img}}_{i,t}\|_2^2}_{\text{image align}} + \underbrace{\|\hat{z}^{\text{txt}}_{i,s} - z^{\text{txt}}_{i,t}\|_2^2}_{\text{text align}}\right) \end{aligned} \tag{16}$$

Students with Knowledge Distillation. For training student models with KD-objectives as specified above, we always use them in conjunction with the standard contrastive loss (either Eq. (6) or Eq. (7)):

$$\mathcal{L}_{\text{dist-only}} = \mathcal{L}_{\text{softmax/sigmoid}} + \lambda_{\text{smax}} \cdot \mathcal{L}_{\text{smax-dist}} + \lambda_{\text{sig}} \cdot \mathcal{L}_{\text{sig-dist}} + \lambda_{\text{fm}} \cdot \mathcal{L}_{\text{fm-dist}} \tag{17}$$

This objective allows us to flexibly combine the different distillation objectives by varying the different loss-weights $\lambda_{\text{smax/sig/fm}}$. By default, we use only the softmax distillation objective with a loss-weight of 2.0, however we perform sweeps over multiple configurations of loss-weights and loss-combinations in our experiments.

Ensemble Teachers. The above distillation setup also easily enables using multiple teacher models in an ensemble for teaching the student. Such an ensemble teacher strategy has been explored in prior SoTA multimodal distillation works [155]. For a

teacher ensemble, the distillation objective simply averages the predicted logits from the different teachers. As an example, an ensemble-softmax-distillation objective would be as follows:

$$\mathcal{L}_{\text{ens-smax-dist}} = -\frac{1}{2bK}\sum_{k=1}^{K}\sum_{i=1}^{b}(\underbrace{\texttt{softmax}(\mathcal{T}_{i,\cdot}^{k})\log\texttt{softmax}(\mathcal{S}_{i,\cdot})}_{\text{image-to-text}}+\underbrace{\texttt{softmax}(\mathcal{T}_{i,\cdot}^{k\,T})\log\texttt{softmax}(\mathcal{S}_{i,\cdot}^{T})}_{\text{text-to-image}}) \quad (18)$$

E. Training Details

Our default configuration follows that of SigLIP [190]. Unless otherwise specified, we train for 3 billion total samples seen, with a batch-size of b=32,678 with the sigmoid contrastive loss (Eq. (7)). The image-encoder takes images resized to (256×256) without any additional augmentations. By default for all our ablation experiments, we use a ViT-S/16 image encoder and a BERT-small text encoder. The image encoder uses global-average pooling (GAP) for the final embedding by default, however for some experiments we also use multi-head attention pooling (MAP) [85, 189]. The text-encoder uses a sentencepiece tokenizer [80] trained on the English-C4 [120] dataset, with a vocabulary size of 32,000. We truncate all text captions to the first 64 tokens. For most experiments, we use an rsqrt learning rate scheduler [189], with a peak learning-rate of 0.001, and linear-warmup and linear-cooldown applied for 10% of total steps. However, for some of our final method comparisons in Tab. 2, we use a cosine learning rate scheduler [131] with a linear-warmup applied for 10% of total steps and peak learning-rate of 0.001. By default, we use a filtering ratio of f=0.8 when using *ACID* sampling, leading to a super-batch-size of B=163,840. We additionally use an ACID sampling temperature of τ=10 for all our experiments. We sweep over λ={0.5,1.0,2.0} for finding the optimal loss-weight for the *Softmax-Distillation* loss (Eq. (3)). We use a weight decay of 0.0001, gradient clipping to a maximum norm of 1.0, and the Adam optimizer with (β_1=0.9, β_2=0.95). All our experiments are conducted with `big_vision` [11] using `jax` [16].

F. About baselines and final ACED models

In this section, we describe the exact architectural details of all the baselines and our ACED models in Tab. 5.

Table 5. **Architectural Details of baselines and *ACED*-F* models.** For each of the baselines and our own *ACED* models, we provide the exact image and text encoder architectures used, the image-resolution used for training, the patch-size for vision-transformer specific encoders, the text sequence-length, training dataset and total compute budget for training in terms of total samples seen.

Method	Samples Seen	Infer. GFlops	Pretraining Dataset	Image Encoder	Text Encoder	Image Resolution	Image Patch Size	Text Seq. Len.
DatologyAI-cls-S/32	2.0B	2.83	Datology-Proprietary	ViT-S/32	BERT-small	224	32	77
DatologyAI-ret-S/32	2.0B	2.83	Datology-Proprietary	ViT-S/32	BERT-small	224	32	77
TinyCLIP-RN30M	15.2B**	6.93	LAION-400M	RN-30M	Custom	224	(-)	77
TinyCLIP-45M/32	15.8B**	3.70	LAION+YFCC-400M	ViT-65M/32	Custom	224	32	77
TinyCLIP-63M/32	15.8B**	5.65	LAION+YFCC-400M	ViT-63M/32	Custom	224	32	77
MobileCLIP-S0	13B*	3.70	DataCompDR-1B	MCi0	MCt	256	(-)	77
ACED-F0	13B	3.30	DataComp-1B	ViT-S/32	BERT-small	256	32	64
DatologyAI-cls-B/32	5.1B	7.39	Datology-Proprietary	ViT-B/32	BERT-base	224	32	77
DatologyAI-ret-B/32	5.1B	7.39	Datology-Proprietary	ViT-B/32	BERT-base	224	32	77
CLIP-KD-RN50	0.5B	9.09	CC-3M+CC-12M	RN-50	BERT-base	224	(-)	77
OpenAI-RN50	13B	9.09	OpenAI-WIT	RN-50	BERT-base	224	(-)	77
OpenAI-CLIP-B/32	13B	7.39	OpenAI-WIT	ViT-B/32	BERT-base	224	32	77
LAION-CLIP-B/32	34B	7.39	LAION-2B	ViT-B/32	BERT-base	224	32	77
DataComp-CLIP-B/32	13B	7.39	DataComp-1B	ViT-B/32	BERT-base	224	32	77
MetaCLIP-CLIP-B/32	13B	7.39	MetaCLIP-2B	ViT-B/32	BERT-base	224	32	77
CLIP-CID-B/32	7.2B	7.39	LAION-225M	ViT-B/32	BERT-base	224	32	77
TinyCLIP-39M/16	20B**	9.48	YFCC-15M	ViT-39M/16	Custom	224	16	77
MobileCLIP-S1	13B*	7.64	DataCompDR-1B	MCi1	BERT-base	256	(-)	77
ACED-F1	13B	7.14	DataComp-1B	ViT-B/32	BERT-small	256	32	64
OpenAI-RN101	13B	12.75	OpenAI-WIT	RN-101	BERT-base	224	(-)	77
MobileCLIP-S2	13B*	10.81	DataCompDR-1B	MCi2	BERT-base	256	(-)	77
ACED-F2	13B	10.29	DataComp-1B	ViT-B/24	BERT-small	240	24	64

G. Comparison with other batch selection methods

In this section, we compare our ACID method with other online batch selection methods in the literature as outlined in Sec. 2. For a fair comparison, we re-implement four batch-selection methods under our setting, namely, Bad-Students [34], Selective-Backprop [70], RHO-Loss [107] and JEST [35]. For this experiment, we pretrain SigLIP models on DataComp-1B [48] for 3B samples seen. For the reference models required by RHO-Loss, JEST and ACID, we use our pretrained WebLI-C++ reference.From Tab. 6, we observe that our ACID method outperforms all the other batch-selection methods by large margins (1.4% better than JEST and 3.2% better than RHO-loss).

Method	IN-val	COCO	27-Avg
IID (baseline)	63.6	42.4	60.1
Softmax-KD	66.1	47.3	62.0
Bad-Students [34]	60.9	49.0	57.8
Sel-BP [70]	63.5	42.7	60.2
RHO-loss [107]	65.9	49.4	62.6
JEST [35]	68.7	53.4	64.4
ACID	**71.0**	**53.6**	**65.8**

Table 6. **ACID outperforms all other online batch selection methods.**

H. Additional Experiments, Ablations and Results

In this section, we provide some additional ablations and more detailed results, augmenting those present in the main paper. We further also include additional baseline comparisons with proprietary models.

H.1. ACIDistill vs. IIDistill scaling

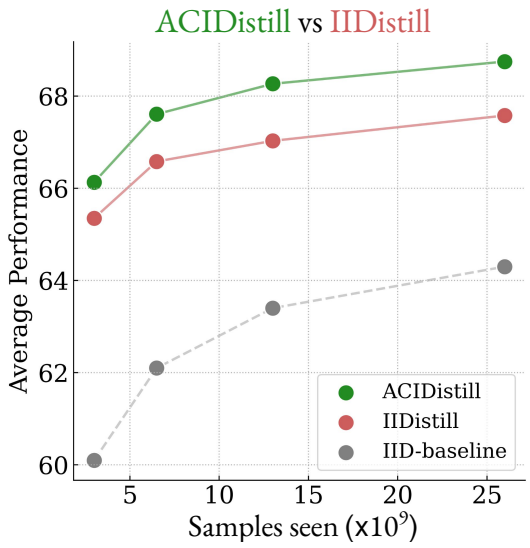

Figure 7. **How to combine ACID and KD in ACED?** The optimal scalable strategy for combining *ACID* and *Softmax-Distillation* is the *ACIDistill* method—where we apply both the contrastive and distillation losses on the *ACID* batch—this is both more performant and training-time efficient than the *IIDistill* scheme.

H.2. Softmax vs Sigmoid Pretraining

We have used SigLIP (sigmoid) pretraining for all our main results because of it's strong performance as a baseline. Here we show that the results are similar with CLIP (softmax) pretraining as well. Overall, the sigmoid variant is more scalable.

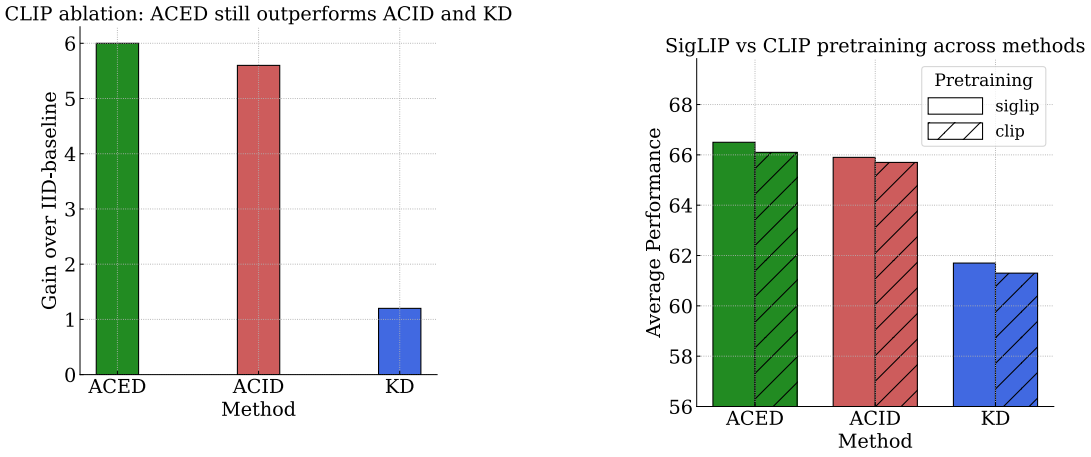

Figure 8. **CLIP vs SigLIP pretraining.** *(left)* Our *ACED* method when applied with CLIP pretraining instead of SigLIP, also further improves over both our *ACID* and *Softmax-KD* approaches. This showcases our methods' generality across pretraining objectives. *(right)* We compare all our methods across SigLIP and CLIP pretraining, and we observe that SigLIP pretraining clearly outperforms the CLIP objective across all the methods, justifying our choice of using it for all our final results.

H.3. ACID vs KD as we scale compute

In Sec. 4.2.2, we demonstrated that our ***ACID*** outperforms distillation methods across a variety of data-, student-size-, and method-configurations. However, all these results were at the 3B samples seen scale. Here, we compare *ACID* and *Softmax-Distillation* as we increase the training compute budget to 6.5B and 13B samples seen scale. Fig. 9 depicts that as we scale up the compute budget, *ACID* still strongly outperforms *Softmax-Distillation*, further signifying the scalability of our method.

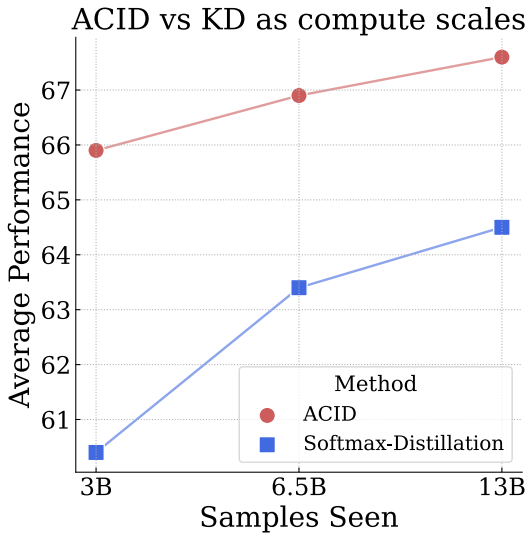

Figure 9. **ACID outperforms Softmax-Distillation across training compute budgets.**

H.4. Full Detailed Results across all 27 *StableEval* Evaluations

Table 7. **Full Detailed Per-Dataset Results of ACED models on the 27 *StableEval* Evaluations.**

	FGVC-Aircrafts	Oxford-Flowers-102	Oxford-IIIT-Pets	Stanford Cars	Food-101	Caltech-101	CIFAR-10	CIFAR-100	Pascal VOC 2007	STL-10	SUN-397	GeoDE	RESISC45	FMoW	ImageNet-V2	ImageNet-A	ImageNet-R	ObjectNet	ImageNet-Val	ImageNet-Sketch	DTD	DollarStreet	Country211	Flickr30k I2T	Flickr30k T2I	COCO I2T	COCO T2I	Average (27)
ACED-F0	18.75	73.41	89.13	79.23	85.41	84.40	93.88	74.38	83.60	97.28	69.12	86.94	64.94	16.46	61.21	33.05	79.15	51.05	68.45	53.37	45.69	43.73	15.09	87.60	71.40	60.80	41.23	64.0
ACED-F1	26.94	79.59	91.31	83.32	89.96	85.24	96.69	81.68	84.39	98.78	73.13	90.49	69.49	23.09	67.80	53.35	87.93	60.24	74.92	61.55	51.54	48.49	20.28	90.30	77.92	64.96	47.27	69.7
ACED-F2	27.00	79.41	92.29	86.48	91.12	83.99	96.03	82.86	85.37	98.85	74.06	91.19	68.84	24.21	70.03	58.64	90.14	63.87	76.90	63.67	50.21	48.42	22.10	91.10	79.46	66.92	49.69	70.9

H.5. Hyperparameter Sensitivity in ACID

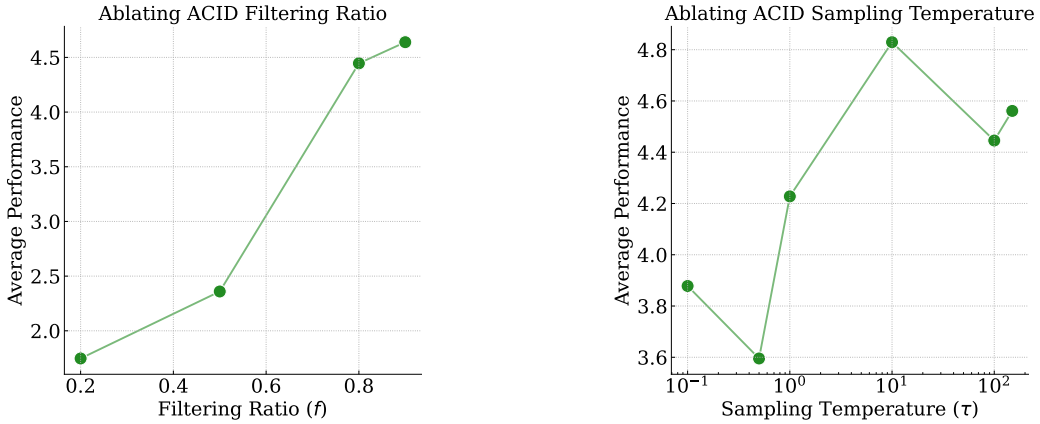

Figure 10. **ACID hyperparameters.** *(left)* We observe that as we keep increasing the filtering ratio, we continue to see improved performance from $f{=}0.2$ to $f{=}0.8$. However, note that these improvements saturate at very high filtering ratios ($f{=}0.9$) due to very aggressive filtering which might lead to insufficient coverage of the entire data distribution. *(right)* We find a sampling temperature $\tau{=}10$ to be optimal across the range of sampling temperatures we tested, trading-off between deterministic top-k sampling (at very high temperatures) vs random sampling (at very low temperatures).

I. Extended Related Works

Multimodal Data Curation. Recent works have emphasised the importance of data quality for pretraining multimodal models [41, 48, 89, 106, 109, 153]. Canonical methods for curating high-quality training data generally involve static offline curation include removing noisy samples [1, 2, 19, 20, 48, 69, 103, 144, 160, 178], rebalancing concept distributions [2, 114, 178], improving quality of text captions [38, 82, 87, 88, 91, 110, 111, 182, 187, 192], and using pretrained data-selector models for filtering samples with low image-text alignment [42, 75, 101, 138, 139, 167–169, 186]. Specifically, it has been shown that offline curation of noisy web-scale data can result in large pretraining efficiency gains [1, 2, 19, 20, 42, 69, 75, 101, 103, 138, 144, 159, 160, 167–169, 178, 186].

However, such static offline curation methods that pre-filter data do not take into account the training dynamics of the current learner model, and hence can suffer at larger scales [53]. Some prior works tackle this by introducing data selection criteria that account for the current state of the learner— Loshchilov and Hutter [100] proposed *online batch selection*, that at each step selects training samples that have the largest learner loss. Further works extended upon this idea by exploring different sample selection criteria, all based on the current learner state [40, 46, 67, 70, 73, 74, 81, 102, 132, 137, 143, 150, 170, 177, 197]. Further, Mindermann et al. [107] introduced the RHO-Loss that considers both current learner state and a pretrained data-selector (reference) model. Further works extended this criterion (termed *learnability scoring*) and scaled it to foundation model training [17, 31, 34, 35, 39, 66]. A key underlying goal of almost all of these prior data curation methods is to improve training efficiency by reducing the number of samples required for pretraining. Owing to this push for training efficiency, most pretrained reference models that are used as *data selectors are typically smaller than the learner models they are used to train* [34, 35, 42]. In fact, Fang et al. [42], Gadre et al. [48], Yu et al. [186] all showed that increasing the reference model size might even be detrimental for training a good learner model.

In this work, we show for the first time that *larger reference models can indeed be used as strong data selectors*, and showcase the conditions under which this simple active data-curation method can be used as an effective distillation strategy for training smaller learner models. Our experiments demonstrate that this can in-fact even outperform standard knowledge distillation strategies that are the most popular methods for compressing big models into smaller, more efficient ones.

Knowledge Distillation. First introduced by Buciluǎ et al. [18] and further popularized by Ba and Caruana [7], Hinton [65], knowledge distillation (KD) is a classic technique for transferring knowledge from a larger model (*teacher*) to another smaller one (*student*), by optimizing the student to match certain outputs (logits, features, intermediate activations etc.) of the teacher model. It has been extensively used for compressing large models into smaller, deployable ones in unimodal tasks like image-classification [12, 22, 25, 43, 47, 63, 113, 128, 149, 158, 165] and language representation learning [5, 55, 76, 95, 134, 146, 179]. Further works have extended KD to use multiple teacher-ensembles [21, 37, 105, 135, 141, 145, 185, 200], different distillation training objectives [68, 92, 122, 147, 151, 175, 196], and progressive multi-stage training schemes [6, 56, 93, 194, 195]. See Gou et al. [52] for a comprehensive survey of KD methods across a range of practical unimodal settings.

However, KD methods in the multimodal foundation model regime are underexplored. Some initial works [29, 44, 99, 166, 171] proposed strategies for efficiently compressing a multimodal teacher for captioning, visual question-answering and video retrieval tasks. Sameni et al. [133] introduced SF-CLIP, a method for improving CLIP pretraining via masked distillation, while Vasu et al. [155] proposed MobileCLIP, exploring downscaling CLIP models for mobile-deployment by using a combination of multi-teacher contrastive-KD, synthetic captions, and data-augmentations. Wu et al. [173] further proposed TinyCLIP—a weight inheritance method combined with an affinity-mimicking strategy for multimodal KD to yield tiny CLIP models. Yang et al. [180] conducted an extensive empirical study (CLIP-KD) into the different objective functions for effectively performing distillation of CLIP models, across different scales. Finally, CLIP-CID [183] uses an image semantic balancing strategy coupled with cluster-instance discrimination for better teacher-to-student knowledge transfer during the KD process. We compare against these methods as baselines for our experimental results in Sec. 4.

Accelerating Knowledge Distillation with Data Selection. There have been prior works attempting to make KD-based pretraining more efficient [140, 142, 188]. Some works [9, 83, 163, 176] have investigated accelerating vanilla KD using active learning in small-scale classification tasks. However, such approaches require a costly iterative process, involving synthetic generation, followed by active sample selection to produce pseudo-labels from a teacher model, thereby limiting their scalability. Another line of work studies data-selection methods for improving KD, typically using uncertainty-based data, logit and feature selection [59, 90, 97, 123, 130, 161, 162, 172, 199], contextual retrieval and sample augmentation from a large data pool [50, 71, 94, 98, 118, 124, 191], or influence-function based sample selection [83, 184]. Contrary to these works, Beyer et al. [12] and Hao et al. [57] suggest that vanilla knowledge distillation provides optimal gains in the "infinite-data regimes". All these prior works however operate primarily in the unimodal image or text classification regime, and none has been scaled up to multimodal foundation model training. We showcase, for the first time, that simple data selection using online batch selection outperforms standard KD for pretraining multimodal models. We further study the

optimal strategies for combining vanilla KD and active data curation to best leverage their complementary strengths.

J. Discussion on training cost vs baselines

In this section, we describe in detail the training costs required by ACID compared to other methods. We first define F_I as the FLOPs-per-iteration of a forward pass of the image encoder of the student model. Similarly, we define F_T as the FLOPs-per-iteration of a forward pass through the student text encoder. We do not consider the cost of the forward passes of teacher/reference models because we can cache their embeddings, as proposed in prior work [155, 155].

Given this, we compute the total FLOPs per normal IID iteration is 3(F_I + F_T). After caching reference embeddings, scoring the super-batch with the student model adds 4(F_I + F_T) for a filtering ratio of 0.8, which gives a total FLOPs / iteration of 7(F_I + F_T) (7/3x overhead compared to IID training).

In Fig. 9, we show that the ACID method trained for 3B examples outperforms Softmax-KD training at 13B examples. Even with the 7/3x overhead, the absolute gains of using ACID are significant compared with additional IID and Softmax-KD training. Further, the main SoTA competition, MobileCLIP [155] has additional forward and backward passes due to an additional synthetic caption batch. This is an overhead of 3(F_I + F_T) - F_I because the initial image forward-pass can be cached for the second batch. This gives a total FLOPs per iteration of 6(F_I + F_T) - F_I. If we compare for example, MobileCLIP-S0 (3.70 inference FLOPs) to ACED-F0 (3.30 inference FLOPs), the training per iteration of MobileCLIP-S0 = 6(2.39 + 1.32) - 2.39 = 19.81 FLOPs and ACED-F0 = 7(3.30) = 23.1 FLOPs. Thus ACED incurs an approx. 15% training overhead compared with MobileCLIP. However, it is worth noting that the methods proposed in Evans et al. [34] for flexible resolution scoring can be used to bring this training budget of ACED down drastically to well below that of MobileCLIP, with little loss in performance. We did not implement this as it has been shown in that prior work. Additionally, although MobileCLIP may have a slight training efficiency, their requirement for generating synthetic captions on new data is far more compute intensive than generating embeddings via our reference-model. Finally, we highlight that in general the main goal of our work (and others) *is to maximize performance at given inference budgets* as it is generally assumed that the training cost of efficient models will be amortized over model lifetime in use.

K. Discussion

Model-based active learning and knowledge-distillation are separate techniques that have traditionally targeted two very different problems. While active learning via online batch selection has focused on improving performance and efficiency of large-scale foundation model pretraining, knowledge-distillation methods seek to achieve highly inference-efficient models by transfer of knowledge from these larger foundation models. In this work, we show theoretically that in fact, active data selection can be cast as a form of implicit knowledge-distillation where the target distribution is now a product of reference (teacher) model probabilities and real labels. With this insight, we develop *ACID*, a powerful method for distilling efficient contrastive multi-modal encoders from larger reference models via online joint-example selection [35]. Notably, this method is a significant and initially counterintuitive departure from traditional active curation paradigms [34, 107] which typically seek reference models that are significantly cheaper in compute compared to the student.

We empirically validate that indeed *ACID* is a strong form of distillation that strictly outperforms traditional forms of knowledge-distillation in training contrastive VLMs. Given the different form of implicit distillation objective in *ACID*, we further demonstrate that this is complementary with traditional softmax-based KD, arriving at a final method, *ACED*, which combines the benefits of each. Using *ACID* we effectively distill models that achieve stronger zero-shot classification and image-text retrieval with cheaper inference FLOPs than prior SoTA methods.

K.1. Limitations

While we see our work as a novel, simple, and scalable paradigm for effective distillation of efficient models, our results are limited in scope to contrastive training of VLMs. Knowledge-distillation can in theory be applied to many problems such as supervised image classification [77], self-supervised learning [23, 54], etc. and it remains to be seen whether our results can be transferred to these domains. Furthermore, while we have shown that we can distill SoTA models that are efficient on a theoretical FLOPs basis, it remains to be seen whether our method can achieve SoTA results when constrained by device latency as is necessary for many edge deployments. We leave it to future work to benchmark our method with SoTA low-latency architectures like FastVIT [154] or MobileNet-V4 [117].