How Does the Downstream Accuracy of Barlow Twins Scale with Pre-training Set Size?

A small-compute characterization with a ViT-Tiny on Tiny-ImageNet subsets

Yan Olerinskiy[1]

Supervisor(s): Jan van Gemert[1]**, Alex Manolache**[1]**, Petter Reijalt**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Yan Olerinskiy
Final project course: CSE3000 Research Project
Thesis committee: Jan van Gemert, Alex Manolache, Petter Reijalt, Mitchell Olsthoorn

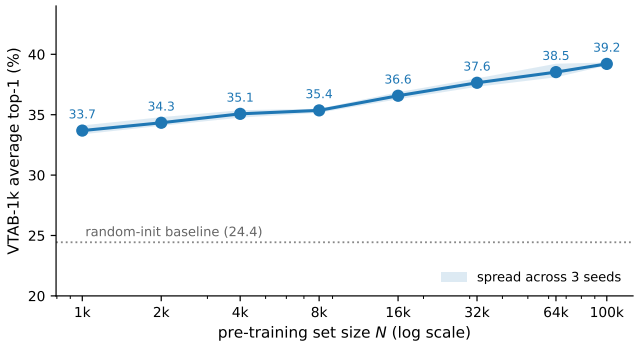

Figure 1: Data efficiency of Barlow Twins (main result). A ViT-Tiny is pre-trained with Barlow Twins on an unlabeled Tiny-ImageNet subset of size N for a fixed 1000 epochs, frozen, and scored by the average accuracy of linear classifiers on the 19 VTAB-1k tasks (mean over three pre-training seeds; the band shows their spread). The average rises with N, from 33.7% at 1k to 39.2% at 100k, well above a 24.4% random-init baseline, with only small gains at the smallest scale. This average hides large differences between task groups (Figure 4).

Abstract

Modern computer vision often reuses a single model, trained once on many images, as a starting point for new tasks. Because labels are expensive, a common way to train such a model is self-supervised learning (SSL), which learns from unlabeled images. SSL normally uses millions of images, and it is unclear how well it works when far fewer are available. We study one SSL method, Barlow Twins, in that case. We pre-train a small vision transformer (5.4M parameters) on parts of Tiny-ImageNet, from 1k to 100k unlabeled images, and train every run for the same 1000 epochs, so the only thing that changes is the amount of data. We then freeze each model and measure how well its features transfer to the 19 VTAB-1k tasks. Pre-training helps at every dataset size: the VTAB-1k average rises from 33.7% with 1k images to 39.2% with 100k, well above a 24.4% untrained baseline. But this average hides large differences between tasks: accuracy on natural-image tasks keeps rising with data, while accuracy on more specialized and structured tasks (medical, satellite, and geometric images) changes little. On the smallest dataset, training too long even lowers accuracy. And as the dataset grows, the checkpoint that scores best on the pre-training data moves further from the one that transfers best. At this small scale, then, the amount of data is not the only thing that matters: the kind of downstream task and the checkpoint we keep matter just as much.

1 Introduction

Modern computer vision often works in two steps: first train a general-purpose model called a foundation model [1] on a large image dataset, then reuse it as a starting point for many different tasks. Because labels are expensive, a common way to train such a model is self-supervised learning (SSL), which learns from the images themselves instead of from labels [4, 25]. SSL normally uses millions of images. Most research groups have far fewer: a team working with medical or satellite images may have only a few thousand. How well SSL works with that little data is much less clear, and that is what we study.

We study one such method, Barlow Twins [25]. It trains a network so that two augmented views of the same image give features that agree, while keeping the feature dimensions uncorrelated; together these two goals keep the representation from collapsing to a trivial solution, without the negative pairs of contrastive methods [4] or the two-network setups of self-distillation methods [2, 11].

We measure data efficiency: how the accuracy on new tasks grows with the number of unlabeled pre-training images N. We use a standard pipeline: we pre-train a network on unlabeled images, freeze it, and train a simple linear classifier (a linear probe) on top for a new task. The probe's accuracy shows how good the frozen features are [4]. We test transfer with VTAB-1k [26], a collection of 19 small classification tasks, and report the average overall accuracy as well as the accuracy for task groups. Changing N gives the curve in Figure 1.

How much data SSL needs has been studied across methods [6, 9, 20], but at much larger scales than ours. We measure it for Barlow Twins on a small network, from 1k to 100k images, so that someone choosing this method for a small dataset knows what to expect.[1] We ask:

> *How does the accuracy of Barlow Twins on new tasks change with the number of unlabeled pre-training images for a small vision transformer, how does this change depend on the kind of downstream task, and how does its representation behave at the smallest dataset sizes?*

Our contributions are:

- We measure the data-efficiency curve of Barlow Twins on the 19-task VTAB-1k average, from 1k to 100k images, for a ViT-Tiny against a randomly initialized baseline. The average rises with dataset size, with only small gains at the smallest scale, but this average hides an uneven split by task group: natural-image tasks keep improving with data, while specialized and structured tasks change little.
- We examine the smallest-data setting and show that training longer lowers validation accuracy: it rises, peaks, and then falls. We check that the features stay spread out rather than collapsing, and read the decline as overfitting, though we do not fully isolate the cause.
- We show that the checkpoint that scores best on the pre-training data is increasingly not the one that transfers best as the dataset grows, so the rule used to pick a checkpoint can hold transfer back.

[1] Code, configurations, and dataset split indices: https://github.com/YanOlerinskiy/barlow-twins-data-efficiency.

These results are averaged over three pre-training seeds.

2 Related Work

Self-supervised learning and Barlow Twins. Most self-supervised methods for images learn by making two augmented views of the same image produce similar representations, while avoiding trivial solutions where every image maps to the same representation. They differ in how they avoid that collapse: contrastive methods such as SimCLR [4] and MoCo-v3 [5] also push apart different images; self-distillation methods such as BYOL [11] and DINO [2] train a student network to match a teacher copy of itself; and masked autoencoders [13] instead reconstruct hidden patches. Barlow Twins [25] takes a different route: it makes the feature dimensions uncorrelated, which avoids trivial solutions without negatives or a second network.

How much data self-supervised learning needs. Several works study this [6, 9, 20], but at much larger scales. For contrastive learning, more data helps little beyond about 500k images [6]. Masked-image methods have been reported to need less data than joint-embedding methods (those that compare two views, like Barlow Twins), though that ranking was measured by fine-tuning and reversed under linear probing [9]. At large scales, joint-embedding methods need almost no augmentation beyond cropping, because augmentation mainly acts as extra data [20]. The smallest dataset Moutakanni et al. test is still 1.3M images [20], which is orders of magnitude above the small-data regime, and none of them measure Barlow Twins on a small network.

Training schedule and checkpoint selection. We compare models trained on different dataset sizes, so the schedule has to be fair across sizes. A popular choice would be to train a cosine decay schedule for a fixed number of epochs, however in our case it can overtrain the smaller datasets. The standard fix is to keep the best-validation checkpoint rather than the last one; but under cosine decay a checkpoint read before the learning rate has decayed has higher loss than a fully decayed one [14, 18], so such a checkpoint would be penalised compared to a checkpoint in the end of the schedule. A constant learning rate removes this penalty: warmup-stable-decay (WSD) schedules [12, 15] hold the rate constant and decay only at the end, matching the final quality of a tuned cosine, so any checkpoint can be selected and then decayed, and checkpoints stay comparable across sizes. Selecting the best checkpoint is commonly done with a validation k-nearest-neighbor classifier on held-out labeled in-domain data, as in DINO [2]; fully label-free alternatives based on representation rank also exist [10].

Collapse and augmentation. A representation can weaken without collapsing completely: in *dimensional collapse* its variance concentrates in a few directions [16], which the effective rank of the feature covariance measures [21]: it is near the full dimension when variance is spread across features, and near 1 when it concentrates in a few. Augmentation is necessary for joint-embedding methods, since it is what makes the two views differ, but a high degree of augmentation is not necessarily better: the augmentation sets what the model becomes invariant to [24], so over-augmenting makes the model invariant to distinctions a downstream task may need, while two views that share too little information lose task-relevant signal [23].

3 Method

We pre-train a small vision transformer with the Barlow Twins objective and evaluate the resulting encoder with a linear probe. This section outlines the approach.

3.1 Encoder and projector

The encoder is a ViT [8]: it splits each image into a grid of square pixel patches and turns them into a single representation vector by pushing them through a stack of transformer layers. The downstream representation corresponds to the value of a special classification (CLS) token. As in the original Barlow Twins paper [25], during pre-training the encoder is followed by a projection head that maps the encoder representation into a higher-dimensional representation, on which the loss is computed. This allows the loss function to remove feature correlations more efficiently, empirically improving performance.

3.2 Barlow Twins loss function

The Barlow Twins loss function makes the projected representations of two augmented views of the same image agree, while trying to keep individual feature dimensions decorrelated. This is done by taking the cross-correlation matrix of projected image representations, computed across a batch. With $z^A, z^B \in \mathbb{R}^{B\times D}$ the batch-normalized projector outputs of the two views (B the batch size, D the projector output dimension), the cross-correlation matrix $\mathcal{C} \in \mathbb{R}^{D\times D}$ has entries

$$\mathcal{C}_{ij} \;=\; \frac{\sum_b z^A_{b,i}\, z^B_{b,j}}{\sqrt{\sum_b \big(z^A_{b,i}\big)^2}\,\sqrt{\sum_b \big(z^B_{b,j}\big)^2}}, \tag{1}$$

where b indexes the batch and i, j index feature dimensions. The loss pushes $\mathcal{C}$ toward the identity matrix,

$$\mathcal{L}_{\mathrm{BT}} \;=\; \sum_i \big(1-\mathcal{C}_{ii}\big)^2 \;+\; \lambda \sum_i \sum_{j\neq i} \mathcal{C}_{ij}^2. \tag{2}$$

The first term makes each feature correlate with itself across the two views, which enforces invariance to augmentations applied in the beginning; the second term makes distinct features uncorrelated, which removes redundancy between features; the λ hyperparameter balances the terms out. A trivial representation cannot satisfy Equation 2, because the diagonal requires perfect correlation across views, so the objective prevents collapse without negative pairs or asymmetric networks [25] that are required by other methods.

3.3 View generation

The two views are produced by the standard two-view augmentation pipeline of BYOL [11]: random resized crop, horizontal flip, color jitter, and random grayscale, with one view additionally solarized. We make two adaptations for low-resolution inputs: the Gaussian blur is left out [4], and the random crop area is increased [23].

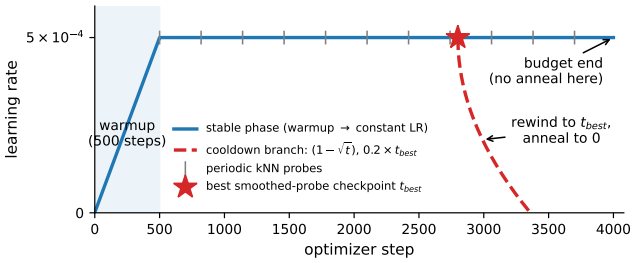

Figure 2: The training schedule. The learning rate warms up, then stays constant while a kNN probe on validation pre-training data periodically runs. After a fixed epoch budget, training rewinds to the best-validation checkpoint and the learning rate decays to zero. No stopping point is fixed in advance, so runs stay comparable across dataset sizes and remain extendable.

3.4 Training schedule and checkpoint selection

For comparing data efficiency, it is important that the training schedule is comparable and equally fair to all dataset sizes. While a cosine decay schedule is commonly used for pre-training, different dataset sizes require different epochs counts to train optimally, and it's not trivial to tune them equally for different datasets. In order to avoid overtraining and make the comparison more fair, we use a warmup-stable-decay (WSD) schedule [12, 15] (Figure 2), which consists of a short linear warmup, then a constant learning rate for a fixed epoch budget and then a short decay to zero. This allows us to take any checkpoint and cool down from it, as opposed to taking the very last checkpoint for a cosine decay schedule.

During the constant phase we score the encoder at a fixed interval with a k-nearest-neighbor classifier on validation pre-training data, and keep the checkpoint with the best smoothed score. We use smoothing because the maximum of a noisy, repeatedly measured signal is biased upward [3].

The final decay restarts from the selected checkpoint rather than the last one, so that a late decline in validation accuracy cannot lower the model we return. The decayed model is the one evaluated downstream.

4 Experimental Setup and Results

4.1 Implementation details

Architecture. As the encoder we use ViT-Tiny/8 [22] with 8×8 patches on 64×64 images: 12 transformer layers, 192-dimensional features, 3 attention heads, 5,388,480 parameters, giving 64 patch tokens and a classification (CLS) token. The vision transformer architecture is now widely adopted [17], and the tiny version allows us to explore its performance in a small-data setup. Similar to the original paper [25], we use a three-layer perceptron projector ($192 \to 1024 \to 1024 \to 1024$, batch normalization and ReLU after the first two layers) but reduced in width from 8192 to 1024. Since this reduction affects the amount of cross-dimensional correlations in Equation 2, the λ is also increased from original $5 \cdot 10^{-3}$ to 10^{-2} to offset the change.

Optimization. We train with AdamW [19] (weight decay 10^{-4}, gradient-norm clip 1.0) at batch size 250, learning rate 5×10^{-4}, with a 500-step linear warmup. AdamW replaces

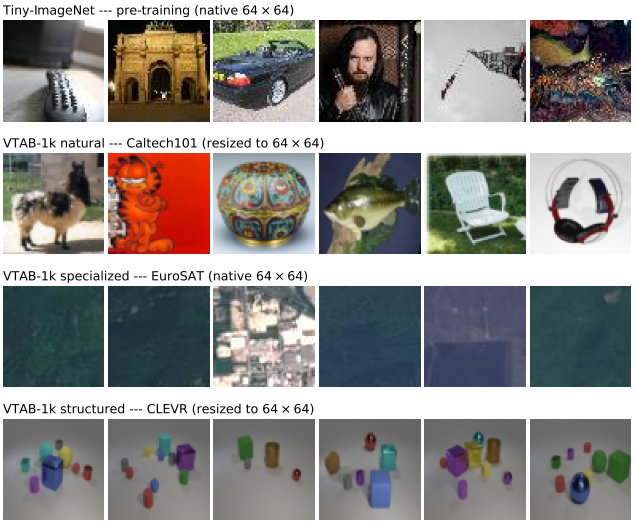

Figure 3: Example inputs at the 64×64 resolution the encoder sees. *Top:* Tiny-ImageNet pre-training images (native 64×64). *Below:* one representative VTAB-1k task per group — Caltech101 (natural), EuroSAT (specialized), and CLEVR-count (structured) — resized to 64×64 for evaluation. The resize does not preserve aspect ratio, so non-square sources (Caltech101, CLEVR) are squashed to square.

the original LARS optimizer, which is meant for large-batch convolutional training; at our batch size, ViT self-supervised methods use AdamW [2, 5, 13]. Batch size is 250, this way it divides every dataset size, so one epoch corresponds to one exposure for every image. The final decay uses a $(1 - \sqrt{t})$ shape over $0.2 \times t_{\text{best}}$ steps (floor 500), where t_{best} is the selected checkpoint's step; this shape and length gave the lowest final loss in the decay-phase ablation of Hägele et al. [12].

Augmentations. Generally, we use the same augmentations and hyperparameters as the original paper [25], with two deviations: the Gaussian blur is left out [4], and the random crop area is increased [23] from 8% to 25% (at 64×64, an 8%-area crop would keep roughly 18×18 pixels, which is about 2×2 patches, meaning two such views would share almost no content).

Pre-training data. Pre-training uses Tiny-ImageNet, a subset of ImageNet [7]: 100,000 images, 200 classes, native 64×64 resolution, and labels are never used (Figure 3). The dataset sizes are $N \in \{1\text{k}, 2\text{k}, 4\text{k}, 8\text{k}, 16\text{k}, 32\text{k}, 64\text{k}, 100\text{k}\}$, drawn as prefixes of shuffled class-balanced lists, so every split is class-balanced and each smaller split is a subset of the larger ones. This way the performance differences come from the added images and not from disjoint datasets. Each split trains for a fixed 1000 epochs, so every split sees every training image 1000 times. The best-validation checkpoint the schedule keeps is reached before the end on every split, so no reported points come from undertrained models.

Validation probe. Checkpoints are scored by a k-nearest-neighbor classifier on the Tiny-ImageNet validation split (10,000 held-out labeled images), split into a 5,000-image reference set and 5,000 queries, 25 images per class. Queries are labeled by their k=20 nearest reference neighbors in CLS-feature space (Euclidean, unweighted). The same probe

Table 1: Learning-rate sweep on the 8k split (single seed). Accuracy is the in-domain kNN top-1 at the selected checkpoint; effective rank (of 192) is measured at the budget end. 5×10^{-4} sits mid-band with near-peak rank; 1.5×10^{-3} collapses.

Learning rate	kNN top-1 (%)	Effective rank
1×10^{-4}	13.9	55
3×10^{-4}	14.7	100
5×10^{-4}	15.3	94
7×10^{-4}	15.8	81
1.5×10^{-3}	3.9	8

also records the effective rank [21] of the features, which serves as a dimensional collapse check.

4.2 Evaluation protocol

We measure transfer with VTAB-1k [26]: 19 classification tasks (7 natural, 4 specialized, 8 structured). For each task we freeze the encoder, extract the 192-dimensional CLS features, and train a fresh linear classifier on the task's 1,000-image training split, evaluating on its full test split, this is the standard VTAB-1k protocol. The classifier trains with SGD (momentum 0.9, no weight decay, batch 256, base learning rate $0.1 \times \text{batch}/256$, cosine-annealed over 90 epochs); images are resized to 64×64 and ImageNet-normalized to match pre-training. The reported transfer score is the mean over the 19 tasks of each task's best test top-1 across the probe epochs.

To measure the baseline accuracy we use a random-initialized encoder, as this allows us to separate the benefits of pre-training from the linear classifier.

Hyperparameter tuning. The pre-training learning rate was tuned using a single-seed sweep on the 8k split (Table 1). Accuracy rises from 10^{-4}, which under-trains, to a usable band at 3–7 $\times 10^{-4}$. It then fails at 1.5×10^{-3}, where the run never improves past warmup and ends with an effective rank of 8 of 192 — the representation has collapsed onto a few dimensions. Within the band, accuracy varies by about one point while effective rank declines past 3×10^{-4}. We use 5×10^{-4}: although 7×10^{-4} scores about half a point higher, we prefer the higher-rank point in the band, because effective rank guards against the collapse failure mode, and 5×10^{-4} is the value DINO uses at batch 256 [2]. We apply this rate to every split without re-tuning.

4.3 How does downstream accuracy scale with pre-training dataset size?

This is the question that someone with a small dataset faces directly: is it worth gathering more images, and what is the benefit? We answer this by pre-training multiple models that differ only in the amount of pre-training data, and studying their average VTAB-1k accuracies.

Figure 1 shows the VTAB-1k average and Table 2 lists every split. Pre-training helps at every N: the smallest split reaches 33.7%, 9.3 points above the 24.4% baseline, and the largest reaches 39.2%, 14.8 points above it. The average rises with N throughout, but with only small gains at the smallest scale: four times the data from 1k to 4k adds 1.4 points, and the average is still rising at 100k.

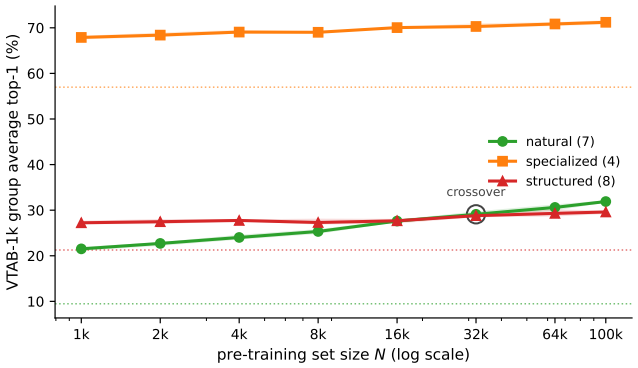

Figure 4: VTAB-1k accuracy split into its three task groups, against pre-training dataset size N (3-seed mean, bands show spread, dotted lines are the per-group random-init baselines). Natural-image accuracy keeps rising with data, specialized accuracy improve slightly and structured accuracy barely changes.

Table 2: VTAB-1k accuracy by task group, per split (3-seed mean, %). The overall 19-task average rises smoothly with N, but the natural group keeps rising while the specialized only improves slightly and structured barely changes. The bottom row is the accuracy of a random-initialized encoder.

N	Overall	Natural	Specialized	Structured
1k	33.7	21.5	67.9	27.2
2k	34.3	22.7	68.4	27.5
4k	35.1	24.0	69.1	27.7
8k	35.4	25.3	69.0	27.3
16k	36.6	27.6	70.0	27.7
32k	37.6	29.1	70.3	28.8
64k	38.5	30.6	70.8	29.3
100k	39.2	31.9	71.2	29.6
baseline	24.4	9.5	57.0	21.3

The average hides where the gains come from. VTAB-1k groups its tasks into natural images, specialized images (medical and satellite), and structured tasks (counting and geometry), and the three scale very differently (Figure 4, Table 2). Natural-image accuracy rises across the whole range, from 21.5% at 1k to 31.9% at 100k (baseline 9.5%), and has not levelled off at 100k. Specialized accuracy jumps from its 57.0% baseline to 67.9% at 1k, then gains only 3.3 more points over the next 100× data. Structured accuracy stays near 27–30% throughout (baseline 21.3%). Natural images start below structured tasks and overtake them around 32k. So more unlabeled data mostly improves natural-image features; the specialized group improves only slightly, and the structured group barely moves with data.

4.4 How does the representation change with continued training on the smallest dataset?

The 1k split is where overfitting is most likely, because the network sees the fewest distinct images. This is the natural place to ask whether more training keeps helping.

On the 1k split the smoothed validation accuracy peaks

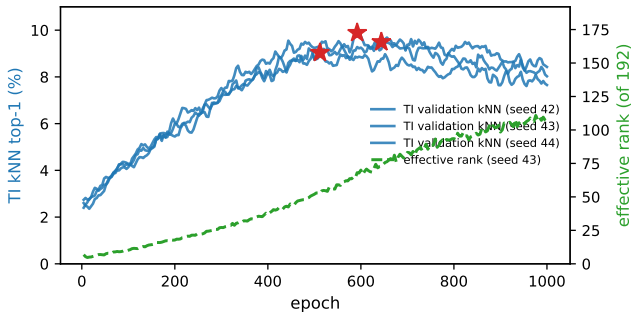

Figure 5: Continued training on the 1k split (all three pre-training seeds): TI validation kNN accuracy (left axis, stars mark the selected checkpoint) and effective rank (right axis, seed 43) against epoch. The validation accuracy peaks near epoch 580 and then falls about 15%, while the effective rank keeps rising to about 106 of 192: we read the late decline as overfitting rather than a collapse onto few dimensions.

near epoch 580 and then falls about 15% by epoch 1000, across all three seeds (Figure 5). The effective rank moves the other way: it rises throughout training, from single digits during warmup to about 106 of 192, and never falls, which is far from the 8 of the collapsed run in Table 1. So the representation does not collapse onto a few dimensions. Instead, we read the decline as overfitting to the 1,000 pre-training images. This is an interpretation: we did not isolate the mechanism, and the constant learning rate is a candidate alternative (Table 1 shows accuracy is sensitive to it). The best checkpoint selection exists specifically to solve this problem, with it the dataset size comparison is more fair.

4.5 How well does pre-training-validation selection track the best checkpoint for transfer?

We select each checkpoint using pre-training validation only, because at selection time we have no downstream labels. The same rule sets every point on the curve, so we ask how much transfer it gives up: we compare it against a checkpoint chosen on each VTAB task's own validation split. That comparison peeks at the downstream tasks, so it is not a usable rule — it is an upper bound on what better checkpoint selection could recover, not an accuracy a label-free rule can reach.

Transfer-aware selection wins at almost every split, and the gap grows with the dataset size: from near zero at the smallest splits to about 1.5 points at 64k–100k (Figure 6, right). The reason is visible within a run (Figure 6, left): the pre-training validation accuracy keeps improving long after the VTAB validation accuracy has peaked, so selecting on it keeps a checkpoint past the point that transfers best, and the distance between the two peaks widens with N. At 64k the VTAB validation accuracy peaks around epoch 250 while the pre-training validation accuracy peaks around epoch 590. The gap is small in absolute terms (near zero to about 1.6 points) and rests on three seeds, so we read the direction — selecting on pre-training validation costs more transfer as N grows — not the exact size.

5 Discussion

Using Barlow Twins, we trained an encoder on various Tiny-ImageNet subset sizes, selected the best checkpoints based on validation accuracy and measured the downstream accuracies with a linear probe. Three results stand out: pre-training helps at every dataset size (9.3 to 14.8 points over the baseline on the 19-task VTAB-1k average), the average rises with N but with only small gains at the smallest scale, and the average hides distinct per-group behavior: natural image accuracy keeps rising with dataset size, while the specialized image accuracy rises only slightly and the structured image accuracy barely moves at all.

What the average hides. It is tempting to read each point of a single data-efficiency curve as the value of that much data. Separating tasks by VTAB group demonstrates the performance in more detail (Figure 4): most of the improvement comes from natural-image tasks, while the specialized tasks improve only slightly and structured groups barely improve. A likely explanation is that Tiny-ImageNet is a subset of ImageNet [7] which consists of natural images, which makes the encoder learn features of natural images and not specialized (satellite, medical) or structured (geometrical) images.

Why augmentations did not change. If augmentations mainly act as extra data [20], then strengthening them is attractive in small-data setups: we increase the amount of images we train the model on without increasing our dataset. Nevertheless, we did not change the augmentations, because, by construction, the Barlow Twins loss function forces the encoder to become invariant to augmentations. A representation made invariant to color can no longer separate a red car from a yellow one, and for an arbitrary downstream task we don't know which distinctions matter in advance.

Compute. All runs use a single NVIDIA RTX 5070 Ti, and any single curve point can be reproduced within a day.

6 Limitations

No convergence claim. Every number is the best-validation checkpoint within a fixed 1000-epoch budget; we do not claim it is the best achievable. The selected checkpoints fall before the budget end (epochs ∼330–930 of 1000), after which validation declines, so the budget runs past the validation peak, not to convergence.

One backbone and one pre-training dataset. All results are limited to ViT-Tiny/8 pre-trained on Tiny-ImageNet. Because we tested only a single backbone, no claim extends to other architectures or sizes, and one pre-training dataset cannot show how the curve depends on the pre-training distribution. In particular, we cannot tell whether the flat specialized and structured groups reflect a ceiling of this encoder or a mismatch between natural-image pre-training and those domains.

No fine-tuning. We measure every encoder with a linear probe on frozen features and do not fine-tune. Fine-tuning asks a different question (how well the features adapt, not how good they are after pre-training). We leave fine-tuning to future work.

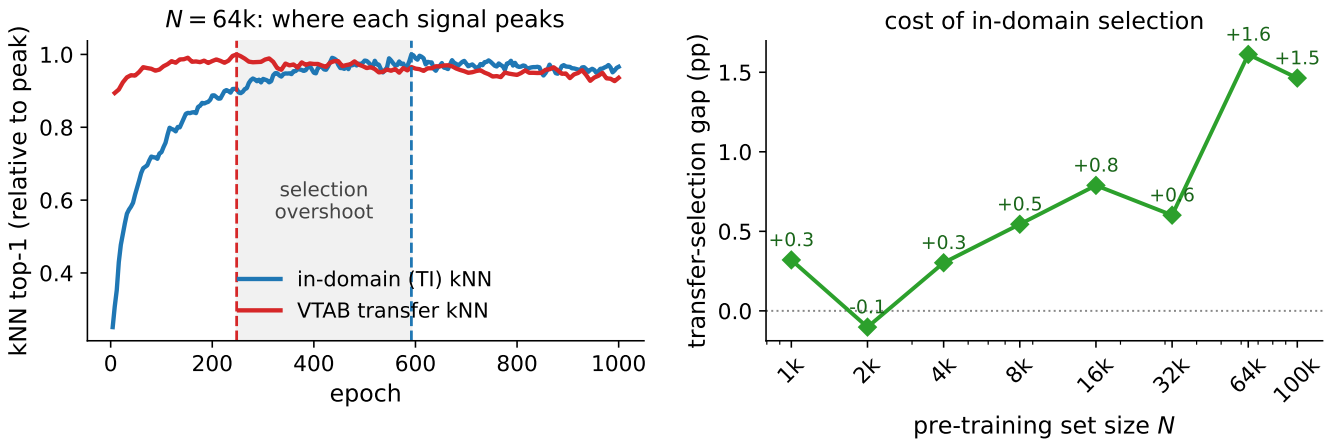

Figure 6: Checkpoint selection. *Left:* on the 64k split, the pre-training validation accuracy (in-domain Tiny-ImageNet kNN, used for selection) peaks far later than the VTAB validation accuracy (transfer kNN), each shown relative to its own peak; selecting on the pre-training validation accuracy keeps a checkpoint well past the transfer peak (shaded). *Right:* the cost, as the VTAB-1k gap between transfer-aware selection (which uses each VTAB task's own validation split) and the pre-training-validation selection we use (3-seed mean). The gap is near zero at the smallest splits and grows with N.

Probe protocol. The VTAB-1k probe uses one fixed recipe for all 19 tasks rather than the per-task hyperparameter sweep of the canonical protocol, and reports the best test top-1 over the probe epochs. This makes the absolute numbers optimistic and not directly comparable to published VTAB-1k results; comparisons across N stay valid because the same recipe is applied to every checkpoint.

Three seeds. The numbers are the mean of three pre-training seeds. The spread across the three seeds is small (within about a point per split) but is a spread and not a confidence interval, so we report it as such and make no claim of statistical significance.

Transferred schedule evidence. Our reasoning relies on properties of WSD schedules (comparability to cosine decay, decay shape and length) that were shown for language-models at larger scales [12, 15]. Their transfer to small scale visual models is an assumption.

7 Responsible Research

Reproducibility. All code, configurations, dataset split indices, evaluation scripts, figure scripts, and the logged metrics of every run are public.[2] Pre-training is reproducible from its seed: the nested splits are prefixes of one published shuffled index list, and all random seeds are fixed; every hyperparameter, including any changes from published defaults, appears in Sections 3 and 4.1. A single consumer GPU reproduces any point on the curve.

Responsible use of AI. We used large language models to help draft this text and to draft and debug our analysis and figure code. All final decisions were made by us. All experimental design, results, and conclusions are our own.

Data and ethics. The study uses public research datasets under their licenses: Tiny-ImageNet (an ImageNet derivative) for pre-training and the VTAB-1k task collection for evaluation. No new data is collected, and no human subjects are involved. These datasets carry the documented biases of their sources, which this study does not audit or correct; the learned encoders are research artifacts for measuring data efficiency, not models intended for deployment.

[2] https://github.com/YanOlerinskiy/barlow-twins-data-efficiency

8 Conclusions and Future Work

We measured how much downstream accuracy improves given additional pre-training data for a ViT-Tiny trained with Barlow Twins, and what happens when the dataset is small. Averaged over three pre-training seeds, pre-training helps at every dataset size: the 19-task VTAB-1k average rises from 33.7% at 1k images to 39.2% at 100k, 9.3 and 14.8 points over a random-initialized encoder, with only small gains at the smallest scale. Looking more deeply at specific VTAB task groups we saw that natural image accuracy keeps rising with more data, while specialized accuracy improves slightly and structured accuracy barely moves. We highlight two further points. First, on the smallest dataset, training past the validation peak lowers accuracy by about 15% with no sign of representational collapse, which we interpret as overfitting. Second, as the amount of data grows, the checkpoint that transfers best diverges more and more from the one that the pre-training validation selects.

To summarize, in small-data setups the training strategy is as important as the amount of training data, and choosing it poorly can measurably reduce transfer accuracy.

For future work, we suggest studying how different pre-training datasets affect downstream performance, as well as adding fine-tuning to the pipeline, which would allow connecting the results of this study to larger data-efficiency studies [9].

References

[1] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. arXiv:2104.14294.

[3] Gavin C. Cawley and Nicola L. C. Talbot. On overfitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. arXiv:2002.05709.

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. arXiv:2104.02057.

[6] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2105.05837.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.11929.

[9] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pretraining? *arXiv preprint arXiv:2112.10740*, 2021.

[10] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2210.02885.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2006.07733.

[12] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.18392.

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2111.06377.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[15] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[16] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2110.09348.

[17] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 2022.

[18] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1608.03983.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1711.05101.

[20] Théo Moutakanni, Maxime Oquab, Marc Szafraniec, Maria Vakalopoulou, and Piotr Bojanowski. You don't need domain-specific data augmentations when scaling self-supervised learning. *arXiv preprint arXiv:2406.09294*, 2024.

[21] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, pages 606–610, 2007.

[22] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. arXiv:2106.10270.

[23] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2005.10243.

[24] Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2008.05659.

[25] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021. arXiv:2103.03230.

[26] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.