

# Benchmarking Image Embeddings for E-Commerce: Evaluating Off-the Shelf Foundation Models, Fine-Tuning Strategies and Practical Trade-offs

Urszula Czerwinska<sup>1</sup> Cenk Bircanoglu<sup>1</sup> Jeremy Chamoux<sup>1</sup>

Adevinta, 22 rue des Jeûneurs, 75002 Paris, France  
`{urszula.czerwinska, cenk.bircanoglu, jeremy.chamoux}@adevinta.com`

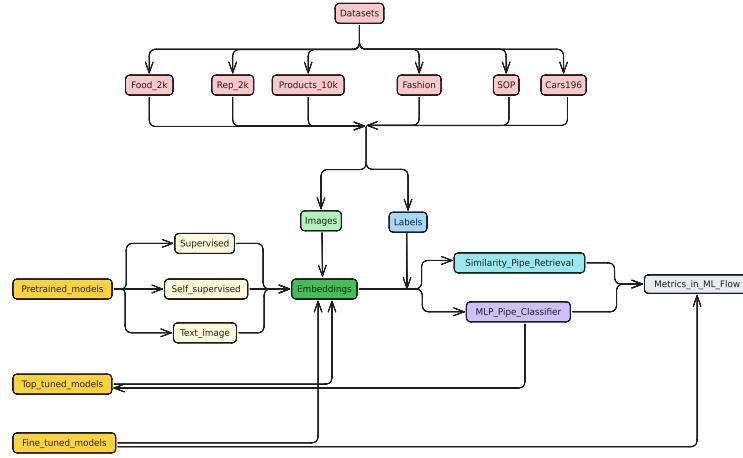
**Abstract.** We benchmark foundation models image embeddings for classification and retrieval in e-Commerce, evaluating their suitability for real-world applications. Our study spans embeddings from pre-trained convolutional and transformer models trained via supervised, self-supervised, and text-image contrastive learning. We assess full fine-tuning and transfer learning (top-tuning) on six diverse e-Commerce datasets: fashion, consumer goods, cars, food, and retail. Results show full fine-tuning consistently performs well, while text-image and self-supervised embeddings can match its performance with less training. While supervised embeddings remain stable across architectures, SSL and contrastive embeddings vary significantly, often benefiting from top-tuning. Top-tuning emerges as an efficient alternative to full fine-tuning, reducing computational costs. We also explore cross-tuning, noting its impact depends on dataset characteristics. Our findings offer practical guidelines for embedding selection and fine-tuning strategies, balancing efficiency and performance.

**Keywords:** foundation models, e-commerce, computer vision, deep learning, pattern recognition

## 1 Introduction

The rapid growth of e-Commerce has intensified the need for accurate and efficient image-based product categorization and retrieval. Machine learning (ML) and computer vision are essential for applications such as search optimization, recommendation systems, and automated product tagging [26]. While foundation models pre-trained image embeddings are widely used for their transferability, selecting the most effective embeddings and fine-tuning strategies for industry-specific applications remains an open challenge [24].

Supervised learning has traditionally been the dominant approach, yet it can be computationally expensive and struggle with generalization across diverse product categories [22]. In contrast, self-supervised learning (SSL) and



**Fig. 1: A high-level illustration of our experimental workflow.** We evaluate pre-trained, fine-tuned, and top-tuned models on six e-Commerce datasets, assessing performance through retrieval. Additionally, pre-trained models undergo classification testing via top-tuning. All metrics are logged in an MLflow dashboard.

contrastive learning offer scalable alternatives by learning meaningful representations without extensive labeled data [3, 27, 13]. However, there is a lack of systematic evaluation comparing how different training modes, backbone architectures (convolutional vs. transformer-based), and fine-tuning strategies impact embedding performance in real-world e-Commerce applications.

To address this gap, we benchmark image embeddings from supervised, self-supervised, and contrastive pretraining foundation methods across six diverse open-source e-Commerce datasets. We evaluate both full fine-tuning and top-tuning, where additional layers are trained on frozen embeddings, to assess their effectiveness in classification and retrieval tasks chapter 1.

Our results show that while full fine-tuning is consistently strong, contrastive text-image models can outperform it, and SSL embeddings can achieve competitive performance with lower computational cost. Additionally, top-tuning significantly enhances all model types while offering a cost-efficient alternative to full fine-tuning. We also analyze cross-tuning—applying top-tuning on a different dataset—highlighting its dataset-dependent effects.

This study provides practical insights for selecting foundation models embeddings and fine-tuning strategies in e-commerce applications, balancing computational efficiency with model performance.

## 2 Related Work

Image embeddings are central to modern computer vision applications, including classification, retrieval, and multimodal tasks. We position our work within key developments in backbone architectures, self-supervised learning (SSL), universal embeddings, fine-tuning strategies, and domain-specific studies, highlighting the novelty of our contributions.

**Backbone Architectures and Representation Learning.** Backbone selection significantly impacts embedding quality. Goldblum et al [6] compare convolutional and transformer-based architectures in various tasks but offer limited information on fine-tuning strategies for domain adaptation. In contrast, we systematically evaluate full fine-tuning, top-tuning, and cross-tuning in the context of e-Commerce.

**Self-Supervised Learning (SSL) Advances.** SSL reduces reliance on labeled data while achieving competitive performance [7, 3, 12]. However, SSL performance varies across tasks [8]. We extend this line of research by demonstrating how top-tuning enhances SSL embeddings, improving their adaptability in e-Commerce scenarios.

**Universal Image Embeddings.** Universal embeddings, foundation model, such as ImageBind [5], aim for broad applicability across vision, audio, and text. While these models offer versatility, their generalization to domain-specific tasks remains underexplored. Our study critically evaluates their performance in specialized e-Commerce applications.

**Supervised Learning as a Benchmark.** Supervised models like ResNet [23] and ViT [4] remain standard baselines but are computationally expensive. We assess supervised embeddings against SSL and contrastive learning approaches, highlighting cost-effective alternatives for e-Commerce tasks with imbalanced or sparse labels.

**Text-Image Embeddings and Multimodal Models.** Contrastive text-image models such as CLIP [13] enable robust multimodal understanding. Prior work by Rashtchian et al [14] explores their semantic properties, but their effectiveness in pure image-to-image retrieval tasks is less examined. We provide a direct comparison of CLIP-based embeddings with traditional vision models, ensuring a fair evaluation by focusing on image retrieval rather than zero-shot or text-to-image tasks.

**Fine-Tuning Strategies for Domain Adaptation.** Fine-tuning techniques, including top-tuning [1, 20] and prompt-based methods [2], offer efficient domain adaptation. FUNGI [15] further explores label-free adaptation. We systematically evaluate fine-tuning and top-tuning, demonstrating the efficiency of top-tuning for e-commerce-specific classification and retrieval.

**Domain-Specific Embedding Studies.** E-Commerce applications demand specialized embedding solutions [9, 17] relying on adaption of a foundational model. While prior work integrates text and image data for recommendations, we focus on systematically benchmarking different embedding paradigms for e-Commerce tasks, addressing dataset diversity and domain-specific nuances.

**Our Contributions.** Existing studies examine backbone architectures, pre-training paradigms, and fine-tuning, but few integrate these perspectives for real-world e-Commerce applications. Unlike prior work, which primarily benchmarks models on generic datasets, we provide a systematic evaluation of supervised, SSL, and contrastive learning embeddings in e-Commerce classification and retrieval tasks. Additionally, we analyze fine-tuning strategies, offering practical guidelines that balance performance and computational efficiency.

### 3 Methods

#### 3.1 Preliminaries

**Image Embeddings in E-Commerce.** Image embeddings encode visual data into compact vector representations, facilitating classification, retrieval, and recommendation tasks.

In our experiments, dataset images were preprocessed according to model requirements, with embeddings stored for downstream evaluation.

#### 3.2 Machine Learning Tasks.

We evaluate embeddings on two key tasks:

**Classification.** Product images are categorized into predefined classes, enabling automated sorting and improved search. We assess classification using a small multi-layer perceptron (MLP) classifier trained on frozen embeddings. This classifier comprises 2-3 fully connected (FC) layers followed by a classification layer. Training consists of (1) Bayesian hyperparameter search over learning rate, momentum, number of FC layers, and optimizer settings (30 epochs, 2 repeats per trial), and (2) model training (1000 epochs with early stopping). We also evaluate cross-tuning, where embeddings from fully fine-tuned models are further adapted to new datasets. For classification, we use standard metrics such as accuracy, precision, recall, and f1-score.

**Retrieval.** Given a query image, retrieval aims to find visually similar products. We index normalized embeddings into a vector database and use L2 distance for nearest-neighbor retrieval. Performance metrics (mMP@5, mR@1 as in Ypsilantis et al [27], MAP, MRR, NDCG) are computed, with  $k = 5$  unless otherwise defined. We evaluate pre-trained, fine-tuned, and top-tuned embeddings, assuming images from the same classification category are similar. This allows direct assessment of embeddings without a classification head.

#### 3.3 Fine-Tuning Techniques

**Full Fine-Tuning.** All model parameters are updated during training on an e-Commerce dataset, allowing deep adaptation but increasing computational cost and overfitting risk. Models start from ImageNet pretraining, undergo standard preprocessing and augmentation, and are trained using a unified parameter set.

Metrics from classification tasks are consolidated, and for retrieval, models are evaluated after removing classification heads.

**Transfer Learning - Top-Tuning.** A computationally efficient approach where pre-trained embeddings remain frozen while a small classifier (2-3 FC layers) is trained on top. This minimizes storage requirements, as embeddings can be stored in flat files, and enables quick adaptation without requiring access to original images.

**Cross-Tuning.** A discovery experiment testing whether fine-tuned embeddings generalize across datasets. A model is fully fine-tuned on dataset A, stripped of its classification layer, and then used to generate embeddings for dataset B, which are evaluated in a retrieval setup. This assesses whether domain adaptation from a related, larger dataset benefits retrieval on a different dataset.

Table 1: Summary of datasets used in the experiments.

Dataset	Domain	Training Size	Test Size	Val Size	Categories
Food2K	Food	620,192	311,859	104,513	2000
Cars196	Cars	8,144	8,041	8,041	196
SOP	Online products	59,551	60,502	60,502	12
Rp2k	Retail products	344,854	39,457	39,457	2384
Product_10k	Retail products	141,931	55,376	55,376	9691
Fashion	Fashion and retail	31,980	4,442	7,996	6

## 4 Datasets

Our study utilizes multiple datasets spanning diverse product categories to ensure a robust evaluation of image embeddings in e-Commerce settings (table 1). A detailed description is provided in appendix A.2.

Fashion serves as a baseline dataset, acting as a technical control due to its relatively small size (40k images) and limited number of classes (six). This makes it the simplest classification task. Product10k, Rep2k, and Food contain a large number of categories, introducing greater complexity. SOP and Cars represent medium-difficulty datasets, balancing dataset size and class diversity.

## 5 Models

To comprehensively assess the impact of backbone architecture, pre-training dataset, and training paradigm, we evaluate a diverse set of deep learning models fig. 2. Our selection spans both supervised and self-supervised learning approaches, as well as contrastive text-image models.

**Supervised Learning Models.** We include well-established architectures widely adopted in both research and industry. These consist of ViT-B, ViT-L [4],

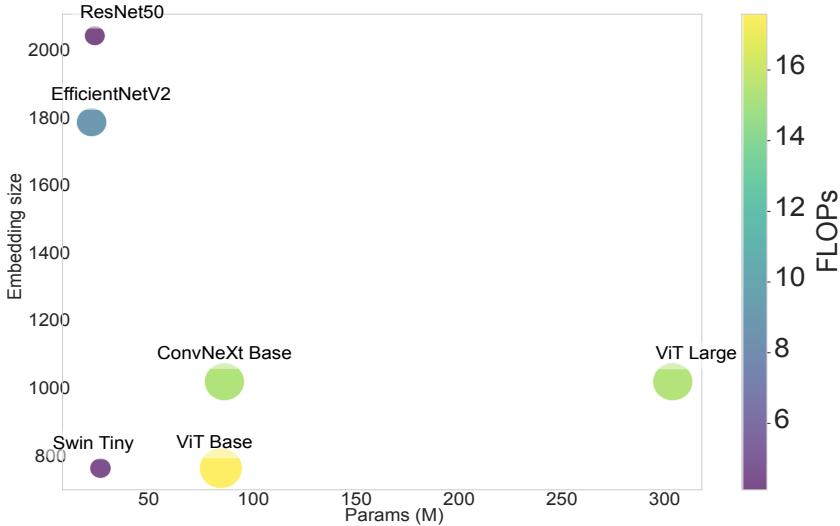


Fig. 2: Models used in this study, showing the relationship between embedding size, FLOPs (B), and parameters (M). DINO, DINoV2, MAWS, MAE, and CLIP share a ViT-B architecture and are represented alongside the vanilla ViT.

ConvNeXt-Base [11], ResNet50 [23], EfficientNetV2 [19], and Swin Transformer [10], all of which have demonstrated strong performance across a range of vision tasks.

**Self-Supervised Learning Models.** We focus on state-of-the-art SSL models that have achieved competitive results in recent studies, including DINO [3], DINoV2 [12], MAE [7], and MAWS [16]. All SSL experiments are conducted using the ViT-B backbone, under the assumption that performance trends observed in ViT-B can be extrapolated to ViT-L.

**Contrastive Text-Image Models.** We evaluate an extensive range of CLIP-style models that differ in pre-training datasets, parameter choices, and architectural variations. These include Meta CLIP [25], EvaCLIP [18], Apple CLIP [21], OpenAI CLIP (ResNet and ViT variants) [13], SigLip [28], and Marqo-B [17], the latter specifically adapted for e-Commerce applications.

This diverse model selection allows us to systematically compare different architectural choices, pre-training strategies, and learning paradigms, offering valuable insights into their relative impact on classification and retrieval performance.

## 6 Experiments and Results

### 6.1 Full Fine-Tuning

For full fine-tuning, models were initialized with ImageNet-pretrained weights and trained following the selected procedure. Some models were subsequently saved without their classification head to generate embeddings, which were evaluated using the retrieval step of the benchmarking pipeline.

**Classification Task.** Following the A2 procedure [23] with a batch size of 512 on four GPUs, results are presented in table 2. The best-performing model was ConvNeXt-Base, achieving 93% accuracy, outperforming the second-best models (ViT-Base and DINO-ResNet50) by 3.6%. Among self-supervised models,

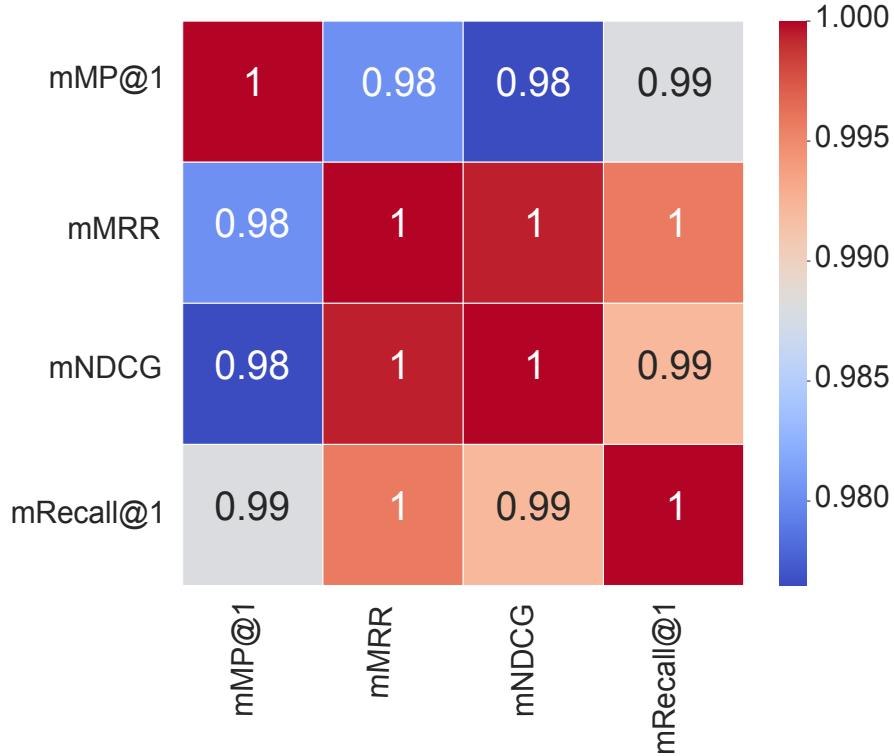


Fig. 3: **Retrieval metric correlation.** All retrieval metrics used in this study are highly correlated.

DINO-ResNet50 and MAE-ViT-B performed competitively, but did not surpass their supervised counterparts. ViT-L was the weakest supervised model, despite having the highest training cost ( $4.6 \times$  that of ViT-B). This suggests that the

dataset size was insufficient for effective ViT-L training. Similarly, MAWS-ViT-B and DINO-ViT-B underperformed, indicating potential limitations in these SSL methods for this setup.

Classification accuracy closely correlates with retrieval performance (mMP@5), which we discuss in detail below fig. 3. Notably, results from the Cars196 dataset (table 2) reveal that self-supervised models exhibit significantly higher variance (0.08) compared to supervised models (0.0009, a  $10\times$  difference), suggesting greater instability in SSL embeddings for this dataset.

Table 2: Performance metrics for different architectures on Cars196 dataset. Best values are bolded.

Architecture	mR@1 (Label 5)	Val set accuracy	Training time (hrs)
ConvNeXt-Base	<b>0.924</b>	<b>0.930</b>	2.67
DINO-ResNet50	—	0.898	<b>1.75</b>
ViT-Base-Patch16	0.886	0.898	2.42
MAE-ViT-Base	—	0.875	2.35
ResNet50	0.826	0.867	1.88
ViT-Large-Patch32	0.797	0.867	8.08
MAWS-ViT-Base	—	0.453	2.37
DINO-ViT-Base	—	0.344	2.45

### Retrieval Task.

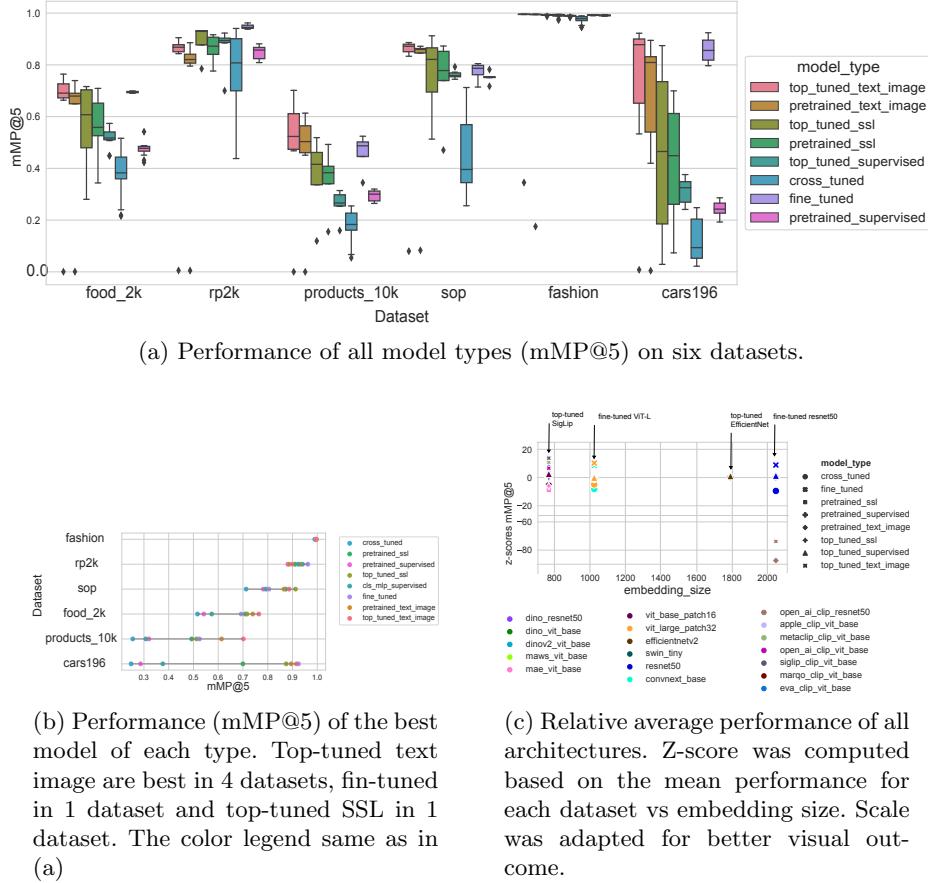
Fine-tuning on the target dataset consistently improves retrieval performance over pre-trained models (fig. 4a), often reaching or approaching state-of-the-art (SOTA) results previously reported in Ypsilantis et al [27]. This highlights the necessity of dataset-specific fine-tuning for optimal retrieval performance.

Among fine-tuned supervised models (fig. 4a), retrieval performance varies between backbones. ConvNeXt-Base achieves the best results on three datasets (Cars196, SOP, Fashion) but performs the worst on Product-10k. ViT-B achieves top performance on two datasets and ranks second on two others, demonstrating strong generalization.

Training time is another key consideration. As shown in table 2, fine-tuning supervised models requires significantly more computational resources than self-supervised models. This variation is largely backbone-dependent, with ViT-L and ConvNeXt training considerably slower than ViT-B or ResNet50. Consequently, training time differences primarily stem from architectural choices rather than the pretraining paradigm.

## 6.2 Pre-trained models

In this section, we analyze the performance of different pre-trained embedding backbones and compare supervised and self-supervised pre-training approaches.



**Fig. 4: All models results.** Retrieval performance comparison for all model types (mMP@5) (a), best models of each type hierarchy per dataset (b) and z-score normalized performance vs embedding size.

Pre-trained embeddings are widely used due to their strong out-of-the-box performance, making them a convenient choice. Here, we evaluate their effectiveness on the retrieval task.

Supervised pre-trained models serve as a strong baseline for retrieval tasks fig. 4a. Performance variance across different backbones is relatively small. Notably, ViT-B achieved the best results in four out of six datasets fig. 5, while also offering an optimal balance between performance and embedding size. Surprisingly, ViT-L underperformed on the `product_10k`, `cars196`, and `SOP` datasets. Additionally, while ConvNext is a strong model when fine-tuned, its out-of-the-box performance is weaker, ranking lowest among pre-trained models on `food2k` and second lowest on `product_10k`.

Table 3: Performance comparison of different models with improvements shown in green.

Model Type	Cars196	Fashion	Food_2k	Prod_10k	Rp2k	SOP	Mean
Pretrained Supervised	0.249	<b>0.991</b>	0.470	<b>0.310</b>	0.855	0.762	0.606
Top-tuned Supervised	<b>0.355</b>	<b>0.991</b>	<b>0.544</b>	0.298	<b>0.908</b>	<b>0.775</b>	<b>0.645</b> ↑3.9
Pretrained SSL	0.533	<b>0.994</b>	0.640	0.428	0.898	0.834	0.721
Top-tuned SSL	<b>0.692</b>	<b>0.994</b>	<b>0.676</b>	<b>0.465</b>	<b>0.931</b>	<b>0.866</b>	<b>0.771</b> ↑5.0
Pretrained Text-image	0.847	0.995	0.695	0.581	0.839	0.860	0.803
Top-tuned Text-image	<b>0.907</b>	<b>0.996</b>	<b>0.739</b>	<b>0.641</b>	<b>0.887</b>	<b>0.882</b>	<b>0.842</b> ↑3.9

As expected, the performance of supervised pre-trained models never surpasses that of fine-tuned ones. Only in the case of the relatively simple **Fashion** dataset does the pre-trained model match fine-tuned performance fig. 4a.

For self-supervised pre-trained models, we observe high performance variance across different backbones, similar to fine-tuned models fig. 4a. Among the SSL models, DINOv2 followed by MAWS achieves the best results, outperforming all other SSL models. Notably, there is little difference in performance between DINO with a ResNet-50 backbone and DINO with a ViT-B backbone, though the ViT-B variant consistently performs better. Compared to fine-tuned models, SSL pre-trained models achieve top performance depending on the dataset. Specifically, on **food2k**, **fashion**, **SOP**, and **product\_10k**, self-supervised pre-trained models outperform fine-tuned ones, with the most significant gains observed on **food2k** and **SOP**.

Finally, text-image pre-trained embeddings demonstrate strong performance on image-to-image retrieval benchmarks, though their performance varies across models fig. 4a. These models achieve the highest performance on **food2k**, **product\_10k**, and **cars196**, outperforming both supervised and self-supervised pre-trained models. Their performance on **rp2k**, **SOP**, and **fashion** is also competitive, making them a solid choice for out-of-the-box image retrieval tasks. Among text-image pre-trained models fig. 5, SigLIP performs best on five out of six datasets, with Apple CLIP outperforming it only on **rp2k**. Apple CLIP also emerges as a strong contender, ranking second-best among text-image models on three datasets. Additionally, on the **product\_10k** dataset, Marqo-B ranks as the second-best model and demonstrates strong performance across **SOP**, **cars196**, **fashion**, **food2k**, and **product\_10k**.

### 6.3 Top-tuned models

We define top-tuning as a transfer learning approach where two or three fully connected linear layers, including a classification layer, are added on top of pre-trained embeddings extracted from the penultimate layer of a pre-trained model. This method is both time- and cost-efficient, making it an accessible solution for data science teams.

Top-tuning proves to be an effective technique across the datasets we tested, particularly when applied to self-supervised model embeddings section 6.3. In the case of supervised pre-trained models, top-tuning improves performance on all datasets except `product_10k`, leading to an average improvement of 3.9%. However, the results still fall short of fully fine-tuned models fig. 4a.

For text-image models, top-tuning consistently improves performance across all datasets section 6.3 and appendix A.1. Since these models already perform well out of the box, the relative improvement is smaller. However, top-tuning allows text-image models to match or surpass supervised fine-tuning on four out of six datasets.

The most significant performance gains are observed when applying top-tuning to self-supervised models (mean 5%) section 6.3 and fig. 6f, fig. 6e. The overall improvement over pre-trained self-supervised models is the highest among all model types, with performance reaching or even exceeding fine-tuned models in some cases—such as on `SOP`. However, the effectiveness of top-tuning varies depending on the model architecture. Specifically, it yields positive results for `dino_vit`, `dinov2_vit`, and `maws` on most datasets (except for `maws` on `SOP`). Conversely, top-tuning negatively impacts performance for `dino_resnet_50` (-6.67% in average) and `mae` (-15.09% in average) across most datasets fig. 6f.

The largest improvement is observed on `cars196`, the smallest and most specialized dataset. This suggests that top-tuning can help pre-trained models specialize for specific tasks. However, its impact should always be compared against the baseline performance of the pre-trained model to determine whether it is a beneficial adaptation.

#### 6.4 Cross-top-tuned models

Fine-tuning on one dataset and applying the model for retrieval on a different dataset typically leads to a significant decline in performance fig. 7 (up to -0.5 mMP@5). However, when the datasets share similar characteristics, cross-top-tuning can have a positive effect (up to 0.1 mMP@5). For example, models fine-tuned on `products_10k` or `cars196` achieve strong results on `rp2k`. Similarly, for `food2k`, a model fine-tuned on `products_10k` attains performance comparable to the best pre-trained model.

## 7 Comparison to prior work

The benchmarking of image embeddings for classification and retrieval has been extensively studied, with prior work comparing pre-training paradigms across diverse domains. Our study builds on this foundation by systematically evaluating fine-tuning strategies, self-supervised learning (SSL), and contrastive text-image models in an e-commerce context, an area with distinct challenges such as high inter-class similarity, long-tailed distributions, and fine-grained retrieval needs.

### 7.1 Embedding benchmarking and performance

The large-scale analysis by Goldblum et al [6] highlights the strengths of convolutional and transformer-based backbones under different pre-training paradigms. Consistent with their findings, we observe that Vision Transformers (ViTs) fine-tuned on domain-specific data consistently outperform convolutional models. However, our results go further by demonstrating that top-tuning SSL and text-image embeddings can yield performance better than or same as full fine-tuning while reducing computational costs.

Unlike prior benchmarks, we show that contrastive text-image embeddings, such as SigLIP, achieve state-of-the-art performance on several retrieval tasks without requiring domain-specific fine-tuning. This suggests that contrastive multimodal pretraining provides robust visual representations even for pure image retrieval, a finding that prior studies have not systematically explored.

### 7.2 Self-Supervised Learning

SSL has gained prominence by reducing dependence on labeled data while achieving competitive results across various tasks [3, 7]. However, prior work has noted the high variability in SSL performance across domains [8]. Our results reinforce this, showing that SSL embeddings exhibit higher variance than supervised models, particularly in retrieval tasks.

Crucially, we find that top-tuning significantly stabilizes SSL embeddings, particularly for DINOv2 and MAWS, narrowing the performance gap with fully fine-tuned supervised models. This supports the argument that SSL models should not be dismissed due to their raw out-of-the-box variability, as lightweight adaptation strategies can enhance their effectiveness with minimal computational overhead.

### 7.3 Fine-Tuning Strategies

Fine-tuning is well-established for domain adaptation, with full fine-tuning often assumed to be necessary for optimal performance [23, 9]. However, its high computational cost limits its scalability. Recent studies explore alternatives such as top-tuning [1] and prompt-based tuning [2], but systematic evaluations in e-commerce contexts remain sparse.

Our results provide a direct comparison of fine-tuning strategies, revealing key trade-offs: (1) **Full fine-tuning** remains the strongest approach but is computationally expensive. (2) **Top-tuning** significantly improves SSL and text-image embeddings, often matching full fine-tuning in retrieval tasks while requiring far fewer resources. (3) **Cross-tuning** shows mixed effectiveness, with gains dependent on dataset similarity, limiting its generalizability.

This suggests that lightweight adaptation strategies are particularly effective for contrastive embeddings, an insight not covered in previous work.

#### 7.4 E-Commerce Context

Most prior studies evaluate embeddings on standard datasets such as ImageNet and COCO [6], which do not fully capture the complexities of e-commerce. Our study fills this gap by benchmarking embeddings across six diverse e-commerce datasets, covering domains such as fashion, retail, food, and automobiles.

Our results reveal that general-purpose pre-trained models do not always perform optimally in e-Commerce settings, particularly in fine-grained retrieval. In contrast, contrastive text-image models—previously optimized for multimodal tasks—demonstrate strong performance in pure image-to-image retrieval, challenging assumptions about their domain specificity. Surprisingly, Marqo-B [17] that was shown to beat its baseline SigLip on zero shot text to image retrieval (labels and categories) does not beat SigLip on image to image retrieval on our benchmark.

Overall, our findings refine prior understanding of embedding selection and adaptation, emphasizing that text-image models and SSL embeddings can achieve state-of-the-art performance in e-Commerce with minimal fine-tuning, significantly reducing computational costs while maintaining retrieval effectiveness.

### 8 Conclusion

Compared to previous studies, our work makes the following key contributions: (1) **Fine-Tuning Strategies in Practice**: We go beyond standard embedding benchmarking by systematically evaluating full fine-tuning, top-tuning, and cross-tuning. Our results provide actionable guidelines for balancing accuracy and computational efficiency in real-world deployments. (2) **Contrastive Text-Image Models for Image Retrieval**: Unlike prior work focused on zero-shot learning, we demonstrate that contrastive models (e.g., SigLIP, Marqo-B) achieve state-of-the-art performance in pure image-to-image retrieval, often outperforming fully fine-tuned supervised models. (3) **Cross-Tuning Analysis**: While most studies focus on direct fine-tuning, we evaluate cross-tuning as a transfer strategy and highlight its dataset-dependent limitations, offering a more nuanced understanding of when adaptation across domains is effective.

Dataset size and granularity strongly influence adaptation strategies, with smaller datasets benefiting the most from top-tuning. Notably, the underperformance of MAE embeddings as frozen features suggests they encode more raw visual information, requiring extensive adaptation for semantic tasks. Despite these insights, limitations remain—our results may not fully generalize beyond e-Commerce, and computational constraints restricted large-scale evaluations. Future research should explore hybrid fine-tuning strategies, automated embedding selection frameworks, and broader multimodal applications in industry. For the practical aspect of usability of embeddings in industrial application, one could investigate the effect of distillation and quantization on embeddings performance.

Overall, our contributions refine the current understanding of embedding selection and adaptation, providing practical guidance for deploying vision models in real-world e-Commerce systems.

**Acknowledgments** We thank **X** for their valuable feedback and insightful suggestions that significantly improved the clarity and quality of this work.

Additionally, we express our gratitude to **Company Name** for providing computational resources and support throughout this research.

## Bibliography

- [1] Alfano PD, Pastore VP, Rosasco L, Odone F (2022) Fine-tuning or top-tuning? transfer learning with pretrained features and fast kernel methods. ArXiv abs/2209.07932, DOI 10.48550/arXiv.2209.07932
- [2] Arango SP, Ferreira F, Kadra A, Hutter F, Grabocka J (2024) Quick-tune: Quickly learning which pretrained model to finetune and how. URL <https://arxiv.org/abs/2306.03828>, 2306.03828
- [3] Caron M, Touvron H, Misra I, Jégou H, Mairal J, Bojanowski P, Joulin A (2021) Emerging properties in self-supervised vision transformers. URL <https://arxiv.org/abs/2104.14294>, 2104.14294
- [4] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: Transformers for image recognition at scale. URL <https://arxiv.org/abs/2010.11929>, 2010.11929
- [5] Girdhar R, El-Nouby A, Liu Z, Singh M, Alwala KV, Joulin A, Misra I (2023) Imagebind one embedding space to bind them all. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp 15,180–15,190, DOI 10.1109/CVPR52729.2023.01457
- [6] Goldblum M, Souri H, Ni R, Shu M, Prabhu V, Somepalli G, Chattopadhyay P, Ibrahim M, Bardes A, Hoffman J, Chellappa R, Wilson AG, Goldstein T (2023) Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. URL <https://arxiv.org/abs/2310.19909>, 2310.19909
- [7] He K, Chen X, Xie S, Li Y, Dollár P, Girshick R (2021) Masked autoencoders are scalable vision learners. URL <https://arxiv.org/abs/2111.06377>, 2111.06377
- [8] Lee S, Lee S, Seong H, Kim E (2023) Revisiting self-similarity: Structural embedding for image retrieval. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp 23,412–23,421, DOI 10.1109/CVPR52729.2023.02242
- [9] Liu C, Hou P, Zeng A, Yu H (2024) Transformer-empowered multi-modal item embedding for enhanced image search in e-commerce. URL <https://arxiv.org/abs/2311.17954>, 2311.17954
- [10] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows. URL <https://arxiv.org/abs/2103.14030>, 2103.14030
- [11] Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. URL <https://arxiv.org/abs/2201.03545>, 2201.03545
- [12] Oquab M, Darct T, Moutakanni T, Vo H, Szafraniec M, Khalidov V, Fernandez P, Haziza D, Massa F, El-Nouby A, Assran M, Ballas N, Galuba W, Howes R, Huang PY, Li SW, Misra I, Rabbat M, Sharma V, Synnaeve G, Xu H, Jegou H, Mairal J, Labatut P, Joulin A, Bojanowski P

- (2024) Dinov2: Learning robust visual features without supervision. URL <https://arxiv.org/abs/2304.07193>, 2304.07193
- [13] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I (2021) Learning transferable visual models from natural language supervision. URL <https://arxiv.org/abs/2103.00020>, 2103.00020
  - [14] Rashtchian C, Herrmann C, Ferng CS, Chakrabarti A, Krishnan D, Sun D, Juan DC, Tomkins A (2023) Substance or style: What does your image embedding know? ArXiv abs/2307.05610, DOI 10.48550/arXiv.2307.05610
  - [15] Simoncini W, Gidaris S, Bursuc A, Asano YM (2024) No train, all gain: Self-supervised gradients improve deep frozen representations. URL <https://arxiv.org/abs/2407.10964>, 2407.10964
  - [16] Singh M, Duval Q, Alwala KV, Fan H, Aggarwal V, Adcock A, Joulin A, Dollár P, Feichtenhofer C, Girshick R, Girdhar R, Misra I (2024) The effectiveness of mae pre-training for billion-scale pretraining. URL <https://arxiv.org/abs/2303.13496>, 2303.13496
  - [17] Sleightholm E (2024) Introducing Marqo Specialized Embedding Models for Ecommerce: Powering Multimodal AI Search — marqo.ai. <https://www.marqo.ai/blog/introducing-marqos-ecommerce-embedding-models>, [Accessed 16-12-2024]
  - [18] Sun Q, Fang Y, Wu L, Wang X, Cao Y (2023) Eva-clip: Improved training techniques for clip at scale. URL <https://arxiv.org/abs/2303.15389>, 2303.15389
  - [19] Tan M, Le QV (2020) Efficientnet: Rethinking model scaling for convolutional neural networks. URL <https://arxiv.org/abs/1905.11946>, 1905.11946
  - [20] Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P (2020) Rethinking few-shot image classification: a good embedding is all you need? ArXiv DOI 10.1007/978-3-030-58568-6\_16
  - [21] Vasu PKA, Pouransari H, Faghri F, Vemulapalli R, Tuzel O (2024) Mobile-clip: Fast image-text models through multi-modal reinforced training. URL <https://arxiv.org/abs/2311.17049>, 2311.17049
  - [22] Villeneuve ASV, Plaisent M (2023) Framework for choosing a supervised machine learning method for classification based on object categories : Classifying subjectivity of online comments by product categories. Proceedings of the 13th International Conference on Advances in Information Technology DOI 10.1145/3628454.3630041
  - [23] Wightman R, Touvron H, Jégou H (2021) Resnet strikes back: An improved training procedure in timm. URL <https://arxiv.org/abs/2110.00476>, 2110.00476
  - [24] Xu D, Yang B (2023) Pretrained embeddings for e-commerce machine learning: When it fails and why? Companion Proceedings of the ACM Web Conference 2023 DOI 10.1145/3543873.3587669
  - [25] Xu H, Xie S, Tan XE, Huang PY, Howes R, Sharma V, Li SW, Ghosh G, Zettlemoyer L, Feichtenhofer C (2024) Demystifying clip data. URL <https://arxiv.org/abs/2309.16671>, 2309.16671

- [26] Ye Y, Huang R, Zeng W (2022) Visatlas: An image-based exploration and query system for large visualization collections via neural image embedding. *IEEE Transactions on Visualization and Computer Graphics* 30:3224–3240, DOI 10.1109/TVCG.2022.3229023
- [27] Ypsilantis NA, Chen K, Cao B, Lipovský M, Dogan-Schönberger P, Makosa G, Bluntschli B, Seyedhosseini M, Chum O, Araujo A (2023) Towards universal image embeddings: A large-scale dataset and challenge for generic image representations. URL <https://arxiv.org/abs/2309.01858>, 2309.01858
- [28] Zhai X, Mustafa B, Kolesnikov A, Beyer L (2023) Sigmoid loss for language image pre-training. URL <https://arxiv.org/abs/2303.15343>, 2303.15343

## A Appendix

### A.1 Appendix I: Supplementary figures

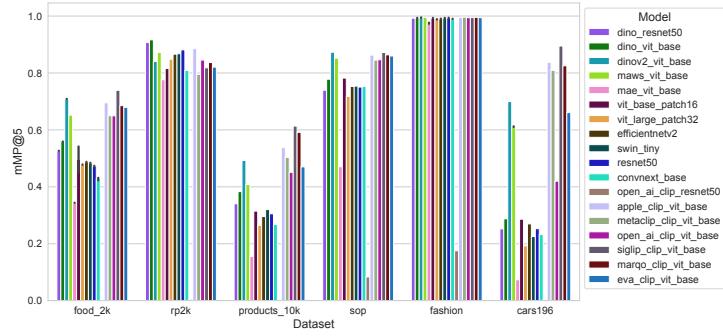


Fig. 5: Detailed results of all pretrained models performance on each dataset.

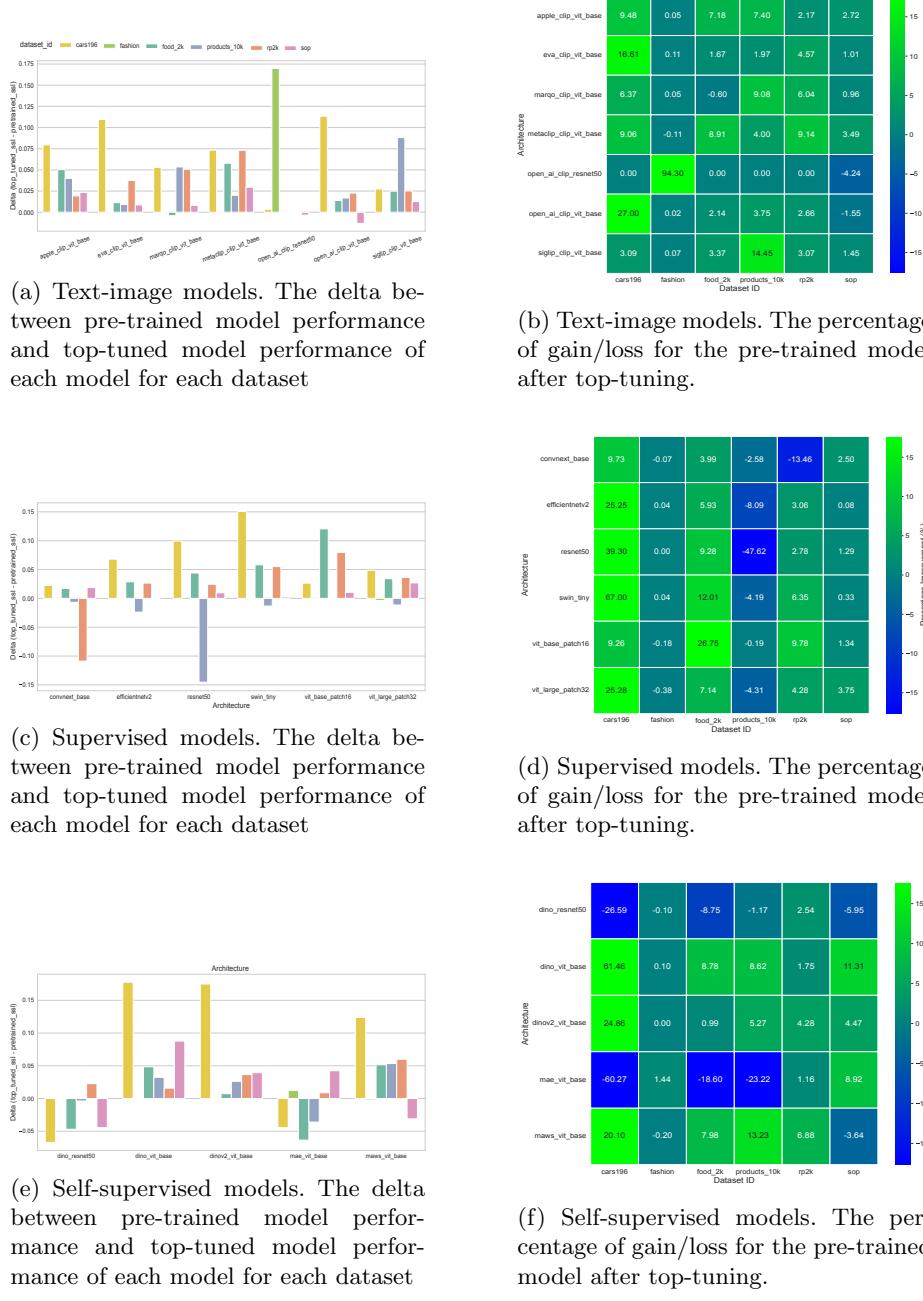


Fig. 6: The detailed analysis of the top-tuning impact on the pretrained models.

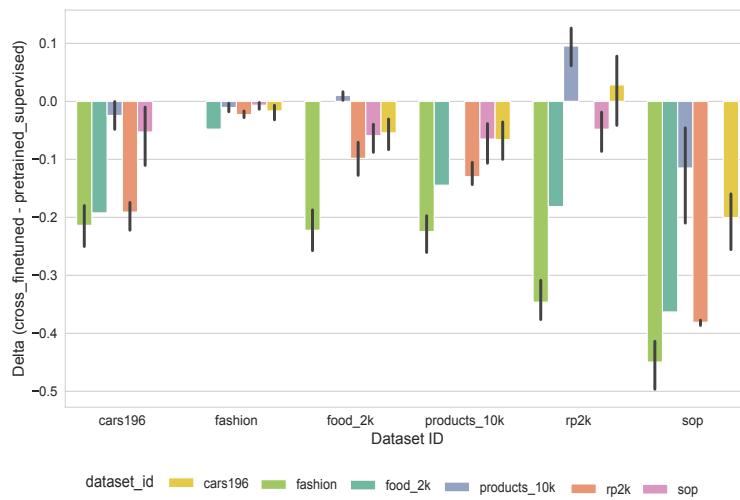


Fig. 7: Cross-tuning. The difference between the cross-tuned and pretrained model in mMP@5 among all tested backbones. X-axes: the top-tuning dataset, color bars correspond to the fine-tuning dataset.

## A.2 Appendix II: Datasets

**Stanford CARS196:** A dataset of 16,185 images across 196 car categories, used for fine-grained classification and visual recognition. Each category corresponds to a distinct car model, with images that vary in viewpoint, lighting, and background.

**Stanford Online Products (SOP):** Contains 120,000+ images of products from 12 categories, designed for metric learning tasks such as product retrieval and visual search. Each product is represented by multiple images from different angles.

**Rp2k:** A large-scale dataset with 2,000 object categories, used for object detection and segmentation tasks. It contains thousands of images per category, supporting research in 3D object recognition.

**Product-10k:** A dataset with 10,000 product categories and images sourced from e-Commerce platforms. It is used for large-scale product recognition and retrieval tasks in real-world conditions.

**Fashion Product Images Dataset:** A dataset containing over 44,000 high-resolution images of fashion products from six categories (e.g., tops, pants, shoes). It includes product images along with additional metadata such as brand, price, and product description, designed for tasks like product classification and retrieval in e-Commerce applications.

**Dataset-Specific Insights** The evaluation across e-Commerce datasets underscores the importance of domain-specific considerations. For instance:

- Smaller datasets with high inter-class separability benefit most from top-tuning.
- Highly granular datasets, such as Product-10k, often require full fine-tuning for optimal performance.

These observations suggest that embedding selection and tuning must be tailored to the characteristics of the target dataset.

## A.3 Appendix III: Implementation details

**Image preprocessing** The images were pre-processed following standard procedures for each model type. Typically provided together with timm library. For models from facebookresearch (torchhub) we applied standard scaling and normalisation. We have tested different mean and variance values for the mae model (given its low performance) but we did not observe significant difference.

**Tuner configuration** For the tuner we used ‘kerastuner’ library. We used bayesian search for following hyperparameters: learning\_rate, number of hidden layers, number of units in hidden layers, activation function, optimiser and clip value. The Tuner runs for 30 epochs for each trial, given 20 trials, each redone twice. As loss function Cross Entropy loss is used. The best model is saved and the training continues from saved checkpoint for the final model keeping best parameter set.

**Full fine-tuning configuration** Selecting the right training parameters is crucial for optimizing model performance. To determine the best configuration, we conducted a series of experiments inspired by Wightman et al [23], identifying the most suitable training procedure for our setup. Given budget and time constraints, we focused on the three top-performing strategies from Wightman et al [23]: A1, A2, and A3.

In our experiments, we tested these three procedures while making two modifications: replacing Binary Cross-Entropy (BCE) with Cross-Entropy (CE) and omitting Stochastic Depth. The decision to use CE was based on the findings in Wightman et al [23], which reported no significant difference between the two loss functions. As for Stochastic Depth, we excluded it to maintain compatibility with pre-trained models, which we intended to use.

Beyond comparing training procedures, we observed that batch size and the number of GPUs significantly impacted performance—particularly for the Cars196 dataset. A batch size of 512, combined with 8 GPUs, produced the best results. Additionally, training a ResNet50 model from scratch yielded noticeably worse performance compared to initializing with pre-trained weights. We attribute this to the limited size of the Cars196 training split, which contains only 8,000 images.

Ultimately, the best results were obtained using the A1 procedure with a batch size of 512 and 4 GPUs. However, the training process took longer than expected, requiring 600 epochs. To balance efficiency and accuracy, we selected A2 with a batch size of 512 and 4 GPUs as our final approach. This configuration resulted in only a 1% drop in validation accuracy compared to the best-performing model while significantly reducing training time.

#### Hardware Specifications

**CPUs and GPUs:** The training was performed on Sagemaker, using 4 Nvidia A10G GPUs. Other experiments were conducted on Kubeflow pipelines using a minimum of 2 CPUs with 8 GB RAM. When necessary, computations were accelerated with 2 GPUs of type g4dn.xlarge, each with 12 GB memory.

**Cluster Configuration:** Kubeflow pipelines were managed with the "unicorn kfp-unicorn" setup, version 2.0.0.

#### Software and Libraries

- **Operating System:** Unix-based systems were used for all experiments.
- **Data Pre-processing:** Pre-processing tasks were performed using PyTorch (`torch==2.1.2`).
- **Model Implementation:** Models were sourced from the `timm` library (`timm==1.0.7`) or `torchhub` (`torchhub==xxx`) and registered in `mlflow` (`mlflow==2.3.0`) registry. All of the models were configured to return embeddings and classification head was removed. Embeddings were generated in Pytorch (`torch==2.1.2`) on a GPU in inference mode.
- **MLP classifier** implemented with Keras v. XXX using Keras Tune v XX
- **Containerization:** Docker image `tensorflow/tensorflow:2.14.0` was employed to ensure consistency and reproducibility for all jobs requiring GPU.

- **Vector Database:** Millvus (`milvus==2.3.0`) was used as the vector database to store embeddings and perform similarity search, interfaced through `pymilvus` (`pymilvus==2.3.6`).

#### A.4 Appendix IV: The Practical Guide to Embedding Choice

This section provides a practical framework for selecting and fine-tuning image embeddings based on our benchmarking results. By addressing common scenarios in e-Commerce and related domains, we offer actionable recommendations for balancing performance, computational efficiency, and task-specific requirements.

**Key Considerations for Embedding Selection** When choosing an embedding model, it is essential to assess the following factors:

- **Dataset Size and Diversity:** Large and diverse datasets benefit from full fine-tuning, whereas smaller datasets often perform well with pre-trained or top-tuned embeddings.
- **Computational Resources:** Resource-intensive models like ViT-L and ConvNeXt require significant training costs. Top-tuning or lighter architectures are more suitable for constrained environments.
- **Task Complexity:** Tasks involving fine-grained classification or retrieval with high inter-class similarity may necessitate full fine-tuning.
- **Label Availability:** When labeled data is scarce, self-supervised learning (SSL) embeddings and text-image embeddings with top-tuning provide a cost-efficient solution.

**Embedding Selection Based on Use Case** Based on our benchmarking analysis, we recommend the following embedding strategies for common e-Commerce tasks:

**Step-by-Step Guidelines** To effectively implement the recommended strategies, follow these structured steps:

**Step 1: Define Task Requirements** Clearly establish the primary objectives (e.g., classification, retrieval) and constraints such as available labeled data, computational budget, and deployment requirements.

#### Step 2: Select the Embedding

- Choose **supervised embeddings** (e.g., ViT, ConvNeXt) for tasks requiring high precision and stability.
- Opt for **self-supervised embeddings** (e.g., DINOv2, MAWS) when adaptability across datasets is a priority.
- Consider **text-image contrastive models** (e.g., CLIP, SigLip, Marqo-B) for multimodal tasks and efficient retrieval.

Table 4: Embedding strategies for common e-Commerce tasks.

Use Case	Recommended Strategy	Rationale
Visual Search and Retrieval	Top-tuned text-image embeddings (e.g., CLIP, SigLip)	Achieves high retrieval accuracy while requiring minimal fine-tuning. Performs well across diverse product categories.
Product Categorization	Fully fine-tuned supervised embeddings (e.g., ViT, ConvNeXt)	Provides stable and high accuracy for structured classification tasks, outperforming SSL embeddings in category-based classification.
Cross-Domain Adaptation	Cross-tuning or top-tuning of text-image embeddings	Enables effective adaptation from one dataset to another. Works best when domains share visual characteristics.
Rapid Prototyping	Pre-trained text-image embeddings (e.g., CLIP, Marqo-B)	Strong zero-shot capabilities, requiring minimal adaptation for fast deployment. Useful for retrieval and tagging tasks.

### Step 3: Decide on Fine-Tuning Strategy

- Use **Full Fine-Tuning** when task-specific adaptation is crucial and computational resources allow.
- Choose **Top-Tuning** for an efficient trade-off between performance and cost, particularly for SSL and text-image embeddings.
- Explore **Cross-Tuning** when domain-specific labeled data is sparse but related datasets are available.

### Step 4: Train and Evaluate

- Train the embeddings using the selected fine-tuning strategy.
- Assess performance with relevant metrics (e.g., accuracy, MAP, Recall@1).
- Iterate by refining hyperparameters and model configurations as needed.

**Step 5: Deploy and Monitor** Deploy the final model in production and continuously monitor its performance. Periodic fine-tuning or retraining may be necessary as new data is collected or task requirements evolve.

**Trade-Offs and Recommendations** Table 5 outlines the trade-offs between fine-tuning strategies, helping to balance performance, efficiency, and domain adaptability.

**Conclusion** This guide provides a structured approach to embedding selection and fine-tuning for e-Commerce applications table 4. Our findings highlight the

Table 5: Trade-offs between fine-tuning strategies.

Strategy	Advantages	Limitations
Full Fine-Tuning	Highest performance, strong task adaptation	Computationally expensive, risk of overfitting.
Top-Tuning	Cost-efficient, significantly improves SSL and text-image models	May not fully match fine-tuned supervised embeddings for classification.
Cross-Tuning	Enables knowledge transfer across domains, useful for text-image models	Performance varies depending on dataset similarity; top-tuning is often more effective.

growing role of text-image embeddings in retrieval and classification, often outperforming supervised models with minimal adaptation. While full fine-tuning remains the strongest approach for domain-specific classification, top-tuning of SSL and contrastive models offers an efficient alternative. Future work should explore hybrid fine-tuning strategies that dynamically adjust between full fine-tuning and top-tuning based on dataset characteristics.

### A.5 Appendix V: Limitations

While our study provides a comprehensive analysis, several limitations should be acknowledged:

**Scope of Datasets and Models** Although our work spans six diverse e-Commerce datasets, there may be unique challenges in other domains that were not captured. Similarly, while we evaluated a broad range of models, certain state-of-the-art architectures (e.g., multi-modal models) were not included.

**Generalizability of Cross-Tuning** The effectiveness of cross-tuning depends heavily on the similarity between source and target datasets. Our findings, while indicative, may not generalize to all cross-domain scenarios. Further exploration of this strategy in diverse domains is necessary.

**Computational Constraints** The resource-intensive nature of some fine-tuning experiments limited the depth of our analysis in certain configurations, particularly for large-scale datasets. Future work could explore additional optimizations to address these constraints.