

TernaryCLIP: Efficiently Compressing Vision-Language Models with Ternary Weights and Distilled Knowledge

Shu-Hao Zhang¹, Wei-Cheng Tang¹, Chen Wu², Peng Hu², Nan Li², Liang-Jie Zhang², Qi Zhang², Shao-Qun Zhang^{1,✉}

¹ State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210063, China

² Microsoft AI, Beijing 100080, China

Abstract

Recent years have witnessed an increasing interest in image-text contrastive modeling, exemplified by models such as Contrastive Language-Image Pretraining (CLIP). In this paper, we propose the TernaryCLIP, a lightweight computational framework that converts connection weights of both vision and text encoders of CLIP into the ternary format, instead of full-precision or floating ones. TernaryCLIP incorporates quantization-aware training and distillation modules, preventing precision degradation and enabling low-cost and high-efficiency computations. Comprehensive experiments demonstrate that TernaryCLIP can achieve up to 99% ternarized weights with 1.58-bit representation, $16.98 \times$ compression ratio, $2.3 \times$ inference acceleration, $16 \times$ storage reduction, $10 \times$ memory optimization, and 60% sparsity while maintaining promising performance on zero-shot image classification and image-text retrieval tasks across 41 commonly used datasets. Our work highlights the feasibility of extreme quantization for large multimodal models, supporting effective and efficient deployment on resource-constrained devices. The model and code can be accessed from [Hugging Face](#) and [GitHub](#).

Key words: Vision-Language Models, CLIP, Ternary Quantization, Knowledge Distillation

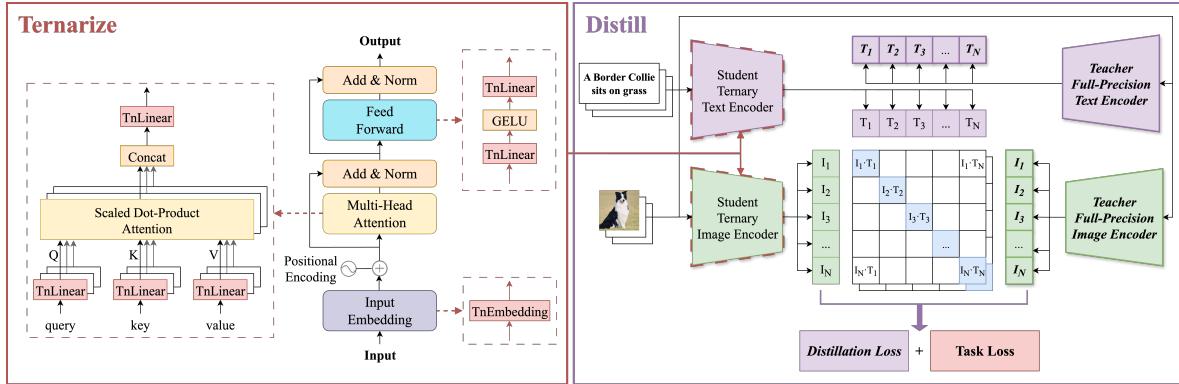


Figure 1: The workflow of the TernaryCLIP framework, comprising two modules: ternarization and distillation.

1. Introduction

Large-scale multimodal models, such as Contrastive Language-Image Pretraining (CLIP) [36] that aligns images and texts in a shared embedding space through contrastive learning, have demonstrated exceptional performance across vision-language tasks, including image classification and image-text retrieval. However, the impressive capabilities of these models come at the cost of significant resource demands [4, 41], contributing to three critical challenges for practical deployment. First, the tremendous parameters result in substantial storage and memory consumption [16]. Second, the full-precision model introduces heavy computational burdens affecting inference latency [42]. Third, the deployment of downstream tasks typically requires massive, domain-specific, labeled datasets [10], such as LAION-400M [40], which incur huge annotation costs and limit practical applicability [5]. These challenges hinder the widespread deployment of large-scale multimodal models in resource-constrained environments such as mobile devices [19]. Consequently, model lightweight techniques have emerged as a prominent research direction to bridge the gap between promising performance and feasible deployment [54].

Quantization has emerged as a promising lightweight technique that converts full-precision weights into lower-bit formats [2, 20, 45]. Specifically, ternary quantization compresses floating-point or 32-bit weights into three discrete values $\{-\Delta, 0, +\Delta\}$, reducing the storage and memory consumption via an ultra-low 1.58-bit format with each weight requiring $\log_2(3) \approx 1.585$ bits in the optimal case. This quantization enables the replacement of complex floating-point matrix multiplications with simple addition and shift operations, thereby improving inference efficiency and reducing latency [1, 30, 53]. However, existing quantization approaches suffer from two critical limitations for multimodal efficient deployment. On the one hand, current research primarily focuses on unimodal architectures such as BERT for natural language processing [50, 52] and ViT for computer vision [24, 32], leaving multimodal models like CLIP largely unexplored. On the other hand, quantization usually causes severe performance degradation that compromises practical applicability [2, 37].

Knowledge distillation has emerged as another lightweight technique that employs the teacher-student framework without massive labeled data [15, 17]. Specifically, distillation resolves the annotation requirement challenge by utilizing soft labels from teacher models to guide student learning, transferring rich representational knowledge without relying on domain-specific labeled datasets [38, 51]. This preserves the zero-shot capabilities of pre-trained models [36], thereby avoiding dependence on extensive labeled datasets and significantly reducing annotation costs. Moreover, distillation addresses training resource consumption by leveraging pre-trained teacher knowledge to substantially reduce student model training time, avoiding the enormous computational overhead of training from scratch [8, 33]. Despite the promising progress, comprehensive distillation strategies for ultra-low-bit multimodal scenarios, such as ternary quantization, remain unexplored [34, 35].

In this paper, we propose the TernaryCLIP by integrating ternary quantization and distillation into large-scale vision-language models. The workflow of our TernaryCLIP is illustrated in Figure 1. The experimental results

indicate substantial improvements in reducing the resource demands. For computational efficiency, TernaryCLIP achieves 1.58-bit weight representation with 99% quantization proportion and 16.98 \times compression ratio, resulting in 2.3 \times inference speedup, 16 \times storage reduction, 10 \times memory optimization, and 60% sparsity. For annotation dependency elimination, TernaryCLIP maintains competitive zero-shot performance across 41 datasets, preserving the generalization capabilities of full-precision models without requiring extensive downstream task-specific annotations. These results demonstrate the effectiveness of our proposed TernaryCLIP, maintaining competitive performance with substantially reduced resource requirements.

To the best of our knowledge, the proposed TernaryCLIP is the first advance that achieves extremely-low compression of weight precision of large-scale vision-language models. Empirical results support the practicability that ultra-low-bit multimodal models are not only feasible but also capable of bridging the gap between performance and practical deployment in resource-constrained environments such as edge computing platforms.

2. Preliminaries

2.1. Introduction to CLIP

The CLIP architecture [36] comprises an image encoder f_i (e.g., ViT) and a text encoder f_t (e.g., BERT), which are trained to maximize cosine similarities between matched image-text pairs while minimizing similarities for unmatched pairs. Given a training dataset $D = (I_i, T_i)_{i=1}^{|D|}$, CLIP learns the cross-modal representations through an InfoNCE-based contrastive loss [43]. TernaryCLIP adopts this contrastive learning framework with the loss function formulated as $\mathcal{L}_{\text{task}} = (\mathcal{L}_{\text{I} \rightarrow \text{T}} + \mathcal{L}_{\text{T} \rightarrow \text{I}})/2$, where $\mathcal{L}_{\text{I} \rightarrow \text{T}} = \text{CrossEntropy}(\text{logits}, \text{labels})$ and $\mathcal{L}_{\text{T} \rightarrow \text{I}} = \text{CrossEntropy}(\text{logits}^T, \text{labels})$. The logits are computed as $\text{logits} = I_e T_e^\top / \tau = f_i(I) f_t(T)^\top / \tau$, representing the scaled cosine similarities between image and text embeddings with a temperature hyperparameter τ .

Inspired by previous distillation methods for CLIP, such as CLIP-KD [47] and ComKD-CLIP [7], TernaryCLIP integrates three complementary distillation strategies: Contrastive Relational Distillation (CRD), Interactive Contrastive Learning (ICL), and Feature Distillation (FD). CRD transfers contrastive knowledge from teacher to student by aligning their cross-modal similarity distributions through KL divergence: $\mathcal{L}_{\text{crd}} = \text{KL}(p_t | p_s) + \text{KL}(q_t | q_s)$, where p_s, p_t and q_s, q_t denote the image-to-text and text-to-image distributions for the student and teacher models, respectively. ICL facilitates cross-modal knowledge transfer by establishing contrastive relationships between student and teacher embeddings across modalities: $\mathcal{L}_{\text{icl}} = (\mathcal{L}_{I_s \rightarrow T_t} + \mathcal{L}_{T_s \rightarrow I_t})/2$, where $\text{logits}_{I_s \rightarrow T_t} = I_{e,s} T_{e,t}^\top / \tau$ and $\text{logits}_{T_s \rightarrow I_t} = T_{e,s} I_{e,t}^\top / \tau$ represent the scaled similarities between student image embeddings and teacher text embeddings, and between student text embeddings and teacher image embeddings, respectively. FD aligns the embedding spaces by minimizing the mean squared error between student and teacher representations: $\mathcal{L}_{\text{fd}} = \text{MSE}(I_{e,s}, I_{e,t}) + \text{MSE}(T_{e,s}, T_{e,t})$, where $\text{MSE}(I_{e,s}, I_{e,t}) = \sum_{i=1}^n (I_{e,s}^{(i)} - I_{e,t}^{(i)})^2 / n$ measures the distance between student and teacher image embeddings, and similarly for text embeddings.

2.2. Related Studies

Model Quantization. Quantization has emerged as a fundamental strategy for compressing neural networks to enable efficient inference. Traditional approaches, such as integer quantization, reduce memory usage and computational costs while maintaining acceptable accuracy [20, 45]. Recent advances have extended to ternary quantization, which represents weights using only three values $\{-1, 0, +1\}$, achieving superior efficiency while preserving model performance [50, 53]. Notable quantization-aware training (QAT) techniques, including TWN [30], TTQ [53], and FATNN [6] incorporate scaling factors to enhance ternary model optimization. The Transformer architecture, which has become essential for modern NLP tasks, has also been successfully quantized through methods such as Q8BERT [50] and TernaryBERT [52], demonstrating the feasibility of low-bit quantization. In computer vision, quantizing Vision Transformers (ViTs) presents unique challenges due to their sensitivity to attention distribution changes, which methods like Q-ViT [24] address through the QAT strategy. In parallel, post-training quantization (PTQ) methods have gained significant attention for training-free advantage under relatively higher bit-width, such as 4-bit and 8-bit quantization. Representative PTQ approaches include AWQ [28], which identifies and preserves salient LLM weights, GPTQ [13] that employs layer-wise quantization with error compensation for LLMs, RepQ-ViT [26] that introduces scale reparameterization for vision transformers, and QwT [14] that achieves quantization without additional training overhead.

Knowledge Distillation for Low-Bit. Pure distillation approaches such as CLIP-KD [47] and ComKD-CLIP [7] have proven effectiveness in enhancing the performance of compressed CLIP models. Concurrently, hybrid methods combining quantization and distillation have emerged to achieve superior compression. For language models, LLM-QAT [31] and BitDistiller [11] demonstrate that knowledge distillation significantly improves quantized model performance. In text embedding applications, TinyBERT [21] and DistilBERT [39] show that distillation from high-precision teachers effectively bridges the accuracy gap in quantized student models.

Multimodal Model Compression. As multimodal models continue to scale up, computational efficiency for edge deployment has become increasingly critical. While pruning, distillation, and quantization have been extensively studied for unimodal architectures [12, 16, 17, 20, 27, 50], ternary quantization for multimodal models remains relatively unexplored compared with unimodal counterparts such as ViT and BERT. Previous works on multimodal model compression such as TinyCLIP [46], CLIP-KD [47], and ComKD-CLIP [7] employ pure distillation for parameter reduction rather than weight quantization.

3. Methodology

In this section, we present the TernaryCLIP, a framework that integrates ternarization and distillation modules to enable efficient vision-language model compression. The connection weights of TernaryCLIP are converted into a ternary format, instead of the full-precision or floating ones of the original CLIP, thereby significantly

compressing the model size. Formally, we employ \mathbf{T} to denote the ternary weights, where each element of \mathbf{T} belongs to $\{-1, 0, 1\}$. Provided the input-output pair (\mathbf{x}, \mathbf{y}) and an apposite loss function $\mathcal{L}(\cdot, \cdot)$, we can build the following optimization in the supervised learning paradigm

$$\mathbf{T}^* = \arg \min_{\mathbf{T}} \mathcal{L}(\mathbf{y}, f(\mathbf{x}, \mathbf{T})) , \quad (1)$$

where $f(\mathbf{x}, \mathbf{T})$ denotes the CLIP equipped with ternary weights. Intuitive approaches to solving Eq. (1) involve converting a pre-trained model from full-precision weights into ternary weights [25] and computing surrogate gradients by adding a collection of full-precision weights \mathbf{W} as latent variables during training [48].

This work proposes an alternative approach, the key idea of which is to align the outputs of the original CLIP and our TernaryCLIP by leveraging knowledge from high-capacity teacher models. Formally, provided the loss function $\mathcal{L}(\cdot, \cdot)$ that conforms to the triangle inequality, the original optimization objective of Eq. (1) can be relaxed by inserting \mathbf{W} as follows

$$\underbrace{\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \mathbf{T}))}_{\textcircled{1}} \leq \underbrace{\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \mathbf{W}))}_{\textcircled{2}} + \underbrace{\mathcal{L}(f(\mathbf{x}, \mathbf{W}), f(\mathbf{x}, \mathbf{T}))}_{\textcircled{3}} . \quad (2)$$

where $\textcircled{1}$ and $\textcircled{2}$ separately describe the training losses led by ternary weight \mathbf{T} and full-precision one \mathbf{W} , and $\textcircled{3}$ is the gap between outputs induced by using \mathbf{W} and \mathbf{T} . Thus, the problem of solving $\textcircled{1}$ in Eq. (1) can be implemented by that of minimizing the sum of $\textcircled{2}$ and $\textcircled{3}$.

The new-build minimization inspires some insights; minimizing $\textcircled{2}$ implies retraining the full-precision weights \mathbf{W} for pursuing high task precision, provided an obtained \mathbf{T} , while minimizing $\textcircled{3}$ searches for ternary weights \mathbf{T} by narrowing the gap induced by using \mathbf{W} and \mathbf{T} , provided an obtained \mathbf{W} . In practice, we solve this minimization by adding the ternarization-aware distillation with a straightforward ternarized module, that is,

$$\mathcal{L}_{\text{ternary}} = \mathcal{L}_{\text{task}}(\mathbf{W}, \mathbf{x}; \mathbf{T}) + \mathcal{L}_{\text{distill}}(\mathbf{W}, \mathbf{x}; \mathbf{T}) , \quad (3)$$

where $\mathcal{L}_{\text{task}}$ indicates contrastive loss and $\mathcal{L}_{\text{distill}}$ denotes the distillation loss that is illustrated in Figure 1 and decomposed as $\mathcal{L}_{\text{distill}} = \lambda_{\text{crd}} \mathcal{L}_{\text{crd}} + \lambda_{\text{icl}} \mathcal{L}_{\text{icl}} + \lambda_{\text{fd}} \mathcal{L}_{\text{fd}}$ with λ_{crd} , λ_{icl} , and λ_{fd} serving as balancing hyperparameters. During training, we employ ternary weights \mathbf{T} for forward propagation while maintaining full-precision weights \mathbf{W} for gradient updates. The task loss $\mathcal{L}_{\text{task}}$ preserves cross-modal alignment capabilities through contrastive learning, while the distillation loss $\mathcal{L}_{\text{distill}}$ transfers knowledge from the teacher model by three complementary mechanisms: contrastive relational distillation, interactive contrastive learning, and feature distillation.

The framework of ternarization-aware distillation enables student models to learn parameters specifically optimized for ternary representation while maintaining the training stability of the full-precision format. Algorithm 1 details the training procedure for TernaryCLIP, establishing a lightweight framework that converts CLIP models to ternary form without sacrificing performance. A key advantage of our framework is its versatility, allowing the weights \mathbf{W} to be randomly initialized for training from scratch, loaded from pre-trained models for fine-tuning, or obtained through distillation for knowledge transfer.

Algorithm 1 Ternarization-Aware Distillation

Input: Full-precision student and teacher CLIP model with weights \mathbf{W} and \mathbf{W}_t , dataset \mathcal{D} , hyperparameter β

Output: Ternarized student CLIP model with weights \mathbf{T} and scaling factor γ

- | | |
|--|--|
| 1. while not converged do | 11. // Backward pass with STE |
| 2. Sample minibatch (I, T) from \mathcal{D} | 12. $\partial \mathcal{L}_{\text{ternary}} / \partial \mathbf{W} \leftarrow \text{STE}(\partial \mathcal{L}_{\text{ternary}} / \partial \mathbf{T})$ |
| 3. // Ternarize weights | 13. // Update full-precision weights |
| 4. $\gamma \leftarrow \beta \sum_{ij} W_{ij} / nm$ | 14. $\mathbf{W} \leftarrow \text{Optimizer}(\mathbf{W}, \partial \mathcal{L}_{\text{ternary}} / \partial \mathbf{W})$ |
| 5. $\mathbf{T} \leftarrow \text{RoundClip}(\mathbf{W}/(\gamma + \epsilon), -1, 1)$ | 15. end while |
| 6. // Forward pass with ternarized weights | 16. // Final ternarization |
| 7. $I_{e,s}, T_{e,s} \leftarrow \text{Encoders}(I, T; \mathbf{T})$ | 17. $\gamma \leftarrow \beta \sum_{ij} W_{ij} / nm$ |
| 8. $I_{e,t}, T_{e,t} \leftarrow \text{Encoders}(I, T; \mathbf{W}_t)$ | 18. $\mathbf{T} \leftarrow \text{RoundClip}(\mathbf{W}/(\gamma + \epsilon), -1, 1)$ |
| 9. $\mathcal{L}_{\text{ternary}} \leftarrow \mathcal{L}_{\text{task+distill}}(I_{e,s}, T_{e,s}, I_{e,t}, T_{e,t})$ | 19. return \mathbf{T}, γ |
-

Here, we employ the straightforward ternarization module as follows

$$\mathbf{T} = \text{RoundClip}\left(\frac{\mathbf{W}}{\gamma + \epsilon}, -1, 1\right) \quad \text{with} \quad \gamma = \frac{\beta}{nm} \sum_{ij} |W_{ij}|,$$

where $\text{RoundClip}(x, a, b) = \max(a, \min(b, \text{round}(x)))$ constrains the rounded values within the range $[a, b]$, γ denotes the adaptive scaling factor, β is the hyperparameter to calibrate quantization threshold tuned by $\arg \min_{\beta} \mathcal{L}_{\text{ternary}}$, and nm indicates the total number of weight elements. This scaling mechanism dynamically adjusts the ternarization threshold based on weight magnitudes, preserving the distributional characteristics of the original weights. The process of ternary conversion involves the non-differentiable $\text{round}(\cdot)$ operation, posing a challenge for gradient-based optimization. To address this, we employ the Straight-Through Estimator (STE) [3], which enables gradient flow through the quantization function, i.e., $\partial \mathcal{L}_{\text{ternary}} / \partial \mathbf{T} \leftarrow \partial \mathcal{L}_{\text{ternary}} / \partial \mathbf{W}$. This module uses ternary weights \mathbf{T} during forward propagation while directing gradients to the full-precision weights \mathbf{W} during backpropagation.

Through joint optimization of task and distillation objectives with ternarized weights, the student model learns parameter distributions inherently suited for ternary quantization while preserving competitive performance. This joint optimization strategy ensures that the compressed model adapts to the constraints of ultra-low-bit ternary representation throughout training, rather than suffering from post-training quantization artifacts. During inference, the model operates exclusively with ternary weights, achieving significant reductions in resource consumption and computational complexity without substantial performance degradation.

4. Experiments

We trained two TernaryCLIP variants and performed extensive evaluations. Q-FFN applies ternary quantization to the embedding layer and feedforward blocks, whereas Q-ALL applies it to additional multi-head attention blocks. Table 1 compares the weight bit-width, quantized components, proportion, and compression ratio across the baseline CLIP model and PTQ variants, including RepQ [26] and QwT [14]. Among these models, TernaryCLIP_Q-ALL achieves the highest quantization proportion of 99% and the largest compression ratio of 16.98 \times . In the rest of the experiments, TernaryCLIP refers to Q-ALL unless specified otherwise.

Methods	Params	Quantization					Comp Ratio \uparrow
		Weight	Emb	MHA	FFN	Prop \uparrow	
Baseline	149.62M	32-bit	\times	\times	\times	0%	1.00 \times
RepQ_Int4	149.62M	4-bit	\times	\checkmark	\checkmark	82.09%	3.55 \times
RepQ+QwT_Int4	159.86M	4-bit	\times	\checkmark	\checkmark	76.83%	3.05 \times
TernaryCLIP_Q-FNN	149.62M	1.58-bit	\checkmark	\times	\checkmark	71.62%	3.13 \times
TernaryCLIP_Q-ALL	149.62M	1.58-bit	\checkmark	\checkmark	\checkmark	99.00 %	16.98\times

Table 1: Comparison of quantization configuration and compression ratio for CLIP models.

4.1. Experimental Setup

Training Configurations. TernaryCLIP variants were trained on Conceptual Captions 12M (CC12M) [5], a large-scale vision-language dataset, with a per-GPU batch size of 384 and a total batch size of 3072 across 8 GPUs. Table 2 presents comprehensive configurations of CLIP models studied here, encompassing pre-trained, distilled, and quantized models. Pre-trained models are ViT-L/14 LAION, ViT-B/16 OpenAI, and LAION. For the distillation configuration, we employ LAION’s CLIP ViT-L/14 as the teacher model to provide transferred knowledge, while using CLIP ViT-B/16 as the student model to achieve the balance between model performance and efficiency. Both ViT-B/16 CLIP-KD and TernaryCLIP are distilled by ViT-L/14 LAION. 4-bit PTQ models contain RepQ and RepQ+QwT quantized from ViT-B/16 OpenAI. TernaryCLIP is the 1.58-bit QAT model.

Hyperparameter Tuning. Systematic hyperparameter tuning was conducted through a two-stage process. First, we performed exploratory training for 32 epochs to identify the selection of promising hyperparameters and assess convergence behavior. Subsequently, we trained the final model for 64 epochs using the AdamW optimizer with these tuned hyperparameters, learning rate of 1×10^{-3} , 1×10^4 warm-up steps, cosine annealing schedule, and weight decay of 0.1. The distillation loss factors are determined as $\lambda_{\text{crd}} = 1.0$, $\lambda_{\text{icl}} = 1.0$, and $\lambda_{\text{fd}} = 2000.0$ to balance different knowledge transfer strategies. Hyperparameter configurations are detailed in Appendix A.

		Models						
		CLIP LAION	CLIP LAION	CLIP OpenAI	CLIP-KD	RepQ	RepQ+QwT	TernaryCLIP
Compression	Pre-Trained	✓	✓	✓	✗	✗	✗	✗
	Quantized	✗	✗	✗	✗	✓	✓	✓
	Distilled	✗	✗	✗	✓	✗	✗	✓
	Parameters	427.6M	149.6M	149.6M	149.6M	149.6M	159.6M	149.6M
Vision	Structure	ViT-L/14	ViT-B/16	ViT-B/16	ViT-B/16	ViT-B/16	ViT-B/16	ViT-B/16
	Image Size	224	224	224	224	224	224	224
	Patch	14	16	16	16	16	16	16
	Layers	24	12	12	12	12	12	12
	Width	1024	768	768	768	768	768	768
Text	Vocab Size	49408	49408	49408	49408	49408	49408	49408
	Context	77	77	77	77	77	77	77
	Layers	12	12	12	12	12	12	12
	Width	768	512	512	512	512	512	512
	Heads	12	12	12	12	12	12	12
Embed Dim		768	512	512	512	512	512	512

Table 2: Model configurations, including training strategy (whether to use distillation or quantization), number of parameters, embedding dimension, vision, and text configurations.

Evaluation Configurations. Comprehensive evaluations were conducted across 41 diverse datasets to assess both effectiveness and efficiency. The evaluations comprise 37 single-label image classification datasets, 1 multi-label image classification dataset, and 3 image-text retrieval datasets. All experiments follow a zero-shot evaluation protocol to demonstrate the generalization capabilities of CLIP models. Performance metrics include Accuracy@1 for single-label image classification, Mean Average Precision (mAP) for multi-label image classification, and Recall@5 for image-text retrieval. To evaluate inference efficiency, we benchmarked compatible models with various precision representations on resource-constrained hardware (Apple M4 Pro ARM CPU), measuring crucial deployment metrics including sparsity, storage footprint, memory consumption, and inference latency, which simulates real-world edge deployment scenarios. Ablation studies examining data augmentation, activation quantization, and self-distillation strategies are presented in Appendix E.

Baseline Comparisons. We compare TernaryCLIP against multiple baselines to demonstrate its effectiveness. LAION CLIP ViT-L/14 serves as the teacher model for both CLIP-KD and our TernaryCLIP variants, enabling the comparison of the distillation baseline. We also include OpenAI and LAION CLIP ViT-B/16 pre-trained models as full-precision baselines for evaluating compressed variants. Additionally, we employ two 4-bit PTQ methods, RepQ [26] and RepQ+QwT [14], on OpenAI CLIP ViT-B/16 as quantization baselines to compare our approach against existing post-training quantization techniques.

Inference Configurations. To ensure reproducibility and fair comparison, we standardize the inference pipeline

across all compatible CLIP models. We adopt the **GCUF** for model storage and leverage the **GGML** framework for efficient inference. For the TernaryCLIP, we implement the TQ1_0 quantization format specifically designed for ternary weights, which enables efficient bit-packing, unpacking, and matrix multiplication operations that minimize both memory overhead and computational latency. Alternative quantization schemes, such as Q4_0, are evaluated in Appendix D. Each benchmark measurement represents the average value of 1,000 independent runs to ensure statistical reliability, with results reported in Table 4.

4.2. Zero-Shot Image Classification Performance

Classification Data Types. The evaluation of TernaryCLIP, alongside other models, involves 37 image classification datasets categorized into three distinct types: 21 natural, 7 specialized, and 9 structured. Natural datasets consist of images depicting real-world objects and scenes encountered by humans, such as those in the *ImageNet series* [9]. Specialized datasets are tailored toward domain-specific applications that require expert knowledge, such as those in the *PatchCamelyon (PCam)* that contains histopathological scans of lymph node sections [44]. Structured datasets focus on spatial relationships, such as *CLEVR*, a synthetic visual question answering dataset [22]. Comprehensive dataset-specific performance results are presented in Appendix C.

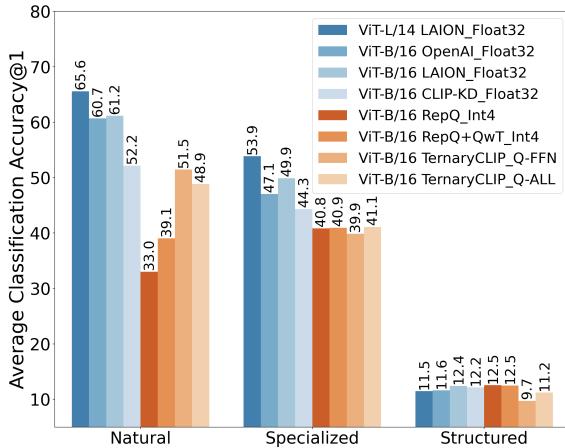


Figure 2: Zero-shot image classification performance (Accuracy@1) on three dataset types and CLIP models. ViT-L/14 or ViT-B/16 indicates which image encoder is used for the structure of CLIP.

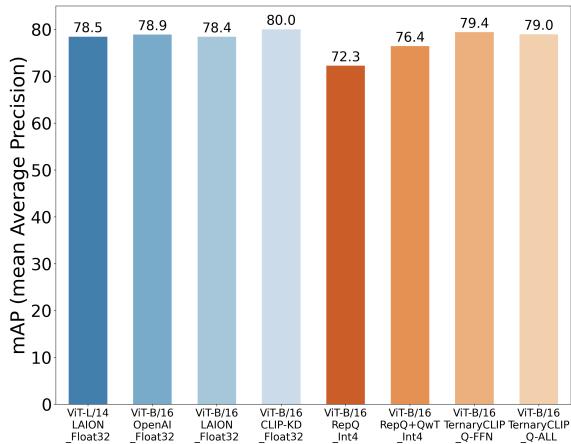


Figure 3: Zero-shot multi-label classification performance (mean average precision) on PASCAL VOC 2007. ViT-L/14 or ViT-B/16 indicates which image encoder is used for the structure of CLIP.

Evaluation of Classification. Figure 2 and Figure 3 demonstrate the effectiveness of TernaryCLIP in preserving performance under weight quantization. The analysis reveals four key findings. First, the performance degradation scales with the full-precision baseline model capacity. Specifically, TernaryCLIP exhibits an average performance decrease of 8.43% relative to the same-sized ViT-B/16 OpenAI and LAION, and 11.96% relative to the larger teacher model ViT-L/14 LAION, both of which are trained from scratch in Figure 2 and

Models	Pre-Trained	Quantized	Distilled	Weight	Average ↑
ViT-L/14 LAION	✓	✗	✗	32-bit	50.20%
ViT-B/16 OpenAI	✓	✗	✗	32-bit	46.17%
ViT-B/16 LAION	✓	✗	✗	32-bit	47.18%
ViT-B/16 CLIP-KD	✗	✗	✓	32-bit	40.95%
ViT-B/16 RepQ	✗	✓	✗	4-bit	29.52%
ViT-B/16 RepQ+QwT	✗	✓	✗	4-bit+32-bit	32.95%
ViT-B/16 TernaryCLIP_Q-FFN	✗	✓	✓	1.58-bit	39.12%
ViT-B/16 TernaryCLIP_Q-ALL	✗	✓	✓	1.58-bit	38.24%

Table 3: Zero-shot image classification average performance (Accuracy@1) on 37 datasets.

Table 3. Despite this degradation, the substantial compression ratio of 16.98 \times makes TernaryCLIP well-suited for edge devices. Second, distillation provides consistent benefits across CLIP variants. Both CLIP-KD and TernaryCLIP achieve over 53% Accuracy@1 on ImageNet-1K [47], resulting in a 17% improvement over the baseline model. This validates the effectiveness of the distillation module in TernaryCLIP. Third, on the multi-label classification dataset PASCAL VOC 2007 in Figure 3, TernaryCLIP variants maintain mAP scores within 1% of full-precision models while outperforming the best PTQ method by 2.6%. Last, TernaryCLIP obtains a remarkable compression ratio with promising accuracy. Compared with the full-precision CLIP-KD, our approach incurs only controllable accuracy loss. Meanwhile, it significantly outperforms 4-bit PTQ methods, including RepQ and RepQ+QwT. In Table 1 and Table 3, TernaryCLIP_Q-ALL achieves a 16.98 \times compression ratio using 1.58-bit weights, with only 2.7% accuracy drop compared with the 32-bit CLIP-KD baseline. Notably, it outperforms RepQ+QwT by 5.29%, despite the latter using higher bit-width representation, 4-bit weights with 32-bit compensation, providing only 3.13 \times compression ratio. These results demonstrate that TernaryCLIP achieves superior compression and competitive performance by integrating ternarization and distillation.

Evaluation of Efficiency. Table 1 presents the efficiency metrics of quantization proportion and compression ratio for quantized models, while TernaryCLIP achieves the highest 99% proportion and 16.98 \times compression ratio. Table 4 presents significant efficiency improvements achieved by TernaryCLIP in terms of sparsity, storage, memory, and latency. TernaryCLIP_Q-ALL achieves 60.88% weight sparsity, enabling further optimization through sparse matrix operations and creating opportunities for additional compression techniques. The ternary weight distributions for sparsity illustration are detailed in Appendix F. The most compressed variant of TernaryCLIP requires only 35.25MB storage, a dramatic reduction from 1630.90MB for the ViT-L/14 Float32 teacher model and 571.60MB for the ViT-B/16 Float32 baseline model, representing 46 \times and 16 \times reductions. This enables deployment on storage-constrained edge devices to host CLIP models. Inference-time memory consumption of model weights and intermediate computations drops from 1690.27MB for the teacher model

Models	Precision	Sparsity \uparrow	Storage (MB) \downarrow	Memory (MB) \downarrow	Latency (ms) \downarrow
CLIP	Float32	0%	1630.90 (100%)	1690.27 (100%)	886.15 (100%)
ViT-L/14	Float16	0%	817.28 (50%)	876.65 (52%)	710.88 (80%)
CLIP	Float32	0%	571.60 (35%)	593.76 (35%)	241.40 (27%)
ViT-B/16	Float16	0%	287.73 (18%)	309.90 (18%)	174.62 (20%)
TernaryCLIP_Q-FFN ViT-B/16	TQ1_0	44.57% (62.27%)	105.00 (6%)	127.18 (8%)	126.05 (14%)
TernaryCLIP_Q-ALL ViT-B/16	TQ1_0	60.88% (61.56%)	35.25 (2%)	57.45 (3%)	106.75 (12%)

Table 4: CLIP model sparsity, storage, memory, and latency on different precisions. For the sparsity x% (y%), x% represents the total sparsity of all parameters, and y% represents the partial sparsity of all ternary parameters.

and 593.76MB for the baseline model to only 57.45MB, achieving 29x and 10x reductions. This breakthrough enables inference with limited RAM on edge devices. TernaryCLIP reduces latency from 886.15ms for the teacher model and 241.40ms for the baseline model to only 106.75ms, delivering 8.3x and 2.3x latency improvements. This acceleration enables real-time applications with strict latency constraints.

TernaryCLIP obtains competitive zero-shot classification performance with aggressive 1.58-bit weight quantization, representing a huge compression ratio compared with full-precision and post-training quantized models, while achieving substantial reductions in storage, memory, and inference costs. These results validate our approach as a practical solution for deploying multimodal models on resource-constrained devices.

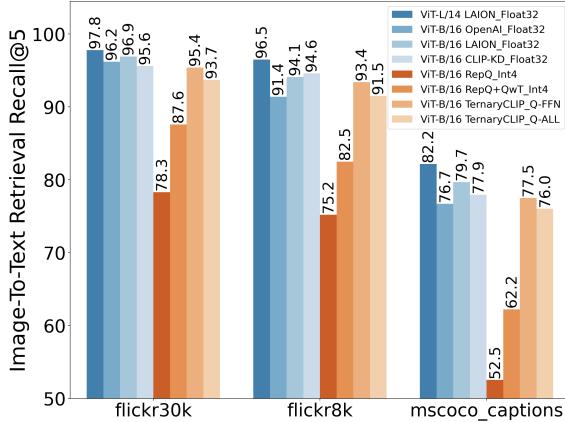


Figure 4: Zero-shot image-to-text retrieval performance (Recall@5) on three datasets and CLIP models. ViT-L/14 or ViT-B/16 indicates which image encoder is used for the structure of CLIP.

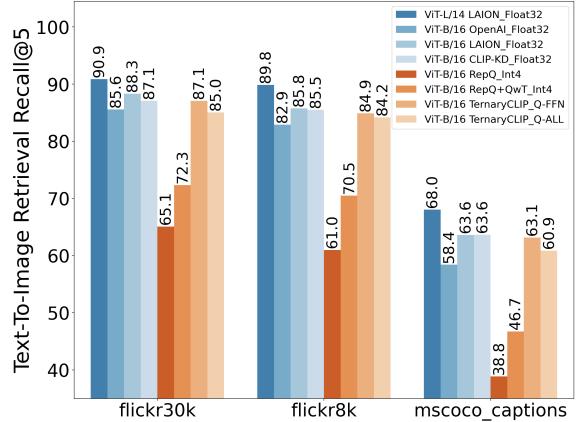


Figure 5: Zero-shot text-to-image retrieval performance (Recall@5) on three datasets and CLIP models. ViT-L/14 or ViT-B/16 indicates which image encoder is used for the structure of CLIP.

4.3. Zero-Shot Image-Text Retrieval Performance

Datasets. To assess zero-shot cross-modal retrieval performance, we select three widely adopted datasets, including Flickr8k [18], Flickr30k [49], and MS COCO [29]. We evaluated two retrieval tasks on these datasets, including image-to-text and text-to-image. Performance is measured using Recall@5 as the evaluation metric.

Evaluation. As shown in Figure 4 and Figure 5, we conducted comprehensive evaluations on various CLIP variants. When evaluated against full-precision pre-trained models (ViT-L/14 LAION, ViT-B/16 OpenAI, and LAION), TernaryCLIP exhibits competitive performance with less than 3% degradation compared with models of equivalent size, and approximately 5% degradation compared with the larger model. Compared with the distillation-only CLIP-KD model under similar training configurations, TernaryCLIP incurs an average performance degradation of only 2.17%. Notably, TernaryCLIP significantly outperforms 4-bit PTQ methods, including RepQ and RepQ+QwT. Leveraging the QAT module, TernaryCLIP with merely 1.58-bit weights achieves an average performance improvement of 11.5% over the best PTQ model, RepQ+QwT employing 4-bit quantized weights and 32-bit compensation weights. These results demonstrate that our approach achieves both the lowest bit-width representation and the highest retrieval performance compared with PTQ methods. Beyond this favorable trade-off between performance and compression, the integration of ternarization-aware training and distillation obtains substantial efficiency improvements, as detailed in Section 4.2.

4.4. Coverage and Costs

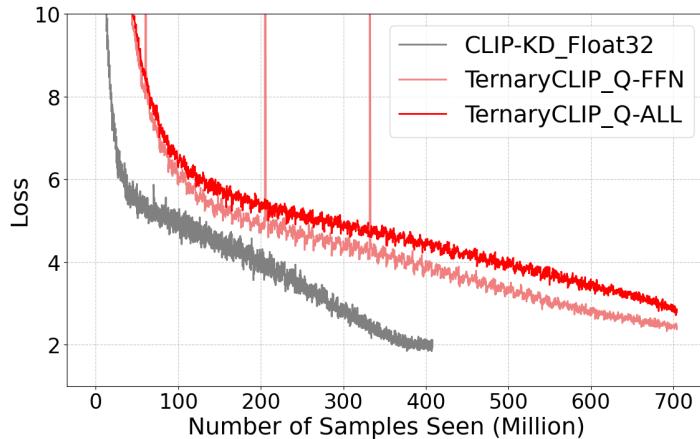


Figure 6: Training total loss curves of models on the number of samples seen: 1) full-precision CLIP-KD, 2) TernaryCLIP_Q-FFN with ternarized EMB+FFN, 3) TernaryCLIP_Q-ALL with ternarized EMB+MHA+FFN.

Convergence. Analysis of the training loss curves in Figure 6 reveals two critical insights of training ternary models. The constituent losses including $\mathcal{L}_{\text{task}}$, \mathcal{L}_{crd} , \mathcal{L}_{icl} , and \mathcal{L}_{fd} are detailed in Appendix B. First, ternary models exhibit oscillatory convergence behavior, in contrast to the smooth trajectory of full-precision training. This

distinction arises from the fundamental difference between continuous and discrete parameter spaces. While full-precision models optimize over a continuous weight space, ternary models operate within a discrete weight space defined by the quantization function $\text{RoundClip}(\cdot)$. Consequently, gradient updates that induce minor changes in full-precision weights can trigger abrupt transitions in ternary weights, resulting in periodic fluctuations in the training loss curves. Second, the quantization proportion directly impacts optimization difficulty. As the proportion of ternarized structures increases, the accumulated quantization error intensifies, requiring a prolonged training budget to achieve convergence and recover from performance degradation. Experimental results demonstrate that, given a sufficient training budget, ternary models can approach the performance of their full-precision counterparts, indicating the viability of ultra-low-bit quantization. This finding suggests that the performance upper bound of ternary models is constrained primarily by training resources rather than the inherent architectural limitation imposed by the compression technique, quantization.

Scaling Law for Ternary Quantization. Inspired by scaling laws in language models [23], we propose a scaling law that characterizes the relationship between a training budget and quantization-induced performance degradation. This law provides a predictive framework for understanding how increased training can mitigate quantization error. Specifically, we formulate the performance gap as

$$\Delta(C, q) = \alpha C^{-\beta} f(q) + c = \alpha C^{-\beta} (1 - q)^{-\gamma} + c, \quad \text{s.t. } \alpha, \beta, \gamma > 0,$$

where $\Delta(C, q)$ represents the performance gap between the quantized and full-precision models, C denotes the training budget, α and β are power law parameters characterizing the compute-performance scaling, $f(q)$ is the quantization penalty function with the quantized proportion $q \in [0, 1]$, γ controls the sensitivity of quantization extent influencing performance degradation, and c is the regularization constant.

Models	Datasets	GPU	Time ↓	Price ↓
ViT-L/14 LAION	LAION-400M	400 * A100	127 hours	44,704\$
ViT-L/14 OpenAI	WIT-400M	256 * V100	288 hours	34,818\$
ViT-B/16 LAION	LAION-400M	176 * A100	61 hours	9,448\$
ViT-B/16 CLIP-KD	CC3M+CC12M	8 * A800	137 hours	998\$
ViT-B/16 TernaryCLIP	CC12M	8 * 6000Ada	129 hours	575\$

Table 5: Comparison of CLIP model training specifications, including Vision Transformer architecture, dataset, GPU requirements, training time, and associated approximate costs.

Training Cost Analysis. To contextualize the computational efficiency, we utilize training resources to compare TernaryCLIP against baseline CLIP models, including OpenAI [36], LAION, and CLIP-KD [47]. While OpenAI does not disclose the detailed training configuration for the ViT-B/16 model, detailed specifications are available for LAION and CLIP-KD, facilitating quantitative comparison. Table 5 summarizes the GPU-hours

and estimated costs for each model, where costs are derived from average pricing on commercial cloud platforms. The results reveal apparent differences in computational requirements across training paradigms. Full-precision models trained from scratch, such as LAION and OpenAI, require extensive GPU resources. In contrast, distillation methods like CLIP-KD and TernaryCLIP achieve competitive performance with significantly reduced training costs. Moreover, TernaryCLIP achieves additional efficiency through ternary quantization.

Overall, the performance trade-off is acceptable given the efficiency gains achieved by TernaryCLIP. Experimental results reveal that under identical model architecture and training configurations, the performance degradation from full-precision CLIP to ternary CLIP is controllable, averaging only 2.7% as shown in Table 3. Meanwhile, TernaryCLIP demonstrates significant improvements in efficiency metrics, including quantization proportion, compression ratio, sparsity, storage, memory, and latency in Table 1 and Table 4. These results demonstrate that our TernaryCLIP achieves a great balance between performance preservation and computational efficiency, making it particularly well-suited for deployment in resource-constrained environments such as edge devices.

5. Conclusions and Prospects

In this paper, we propose the TernaryCLIP, which compresses both the vision and text encoders of CLIP into the ternary format to meet the demands for reducing memory consumption, storage footprint, inference latency, and annotation costs for downstream tasks. By integrating ternarization and distillation modules, TernaryCLIP successfully compresses model parameters to an ultra-low 1.58-bit weight precision while achieving up to 99% ternary weight, 16.98 \times compression ratio, 2.3 \times inference acceleration, 16 \times storage reduction, 10 \times memory optimization, and 60% sparsity without compromising the model’s zero-shot capabilities evaluated on comprehensive cross-modal understanding tasks. Experimental results across 41 benchmarks demonstrate the effectiveness and efficiency of TernaryCLIP, supporting deployment on resource-constrained edge devices. Therefore, TernaryCLIP establishes a new paradigm for lightweight vision-language models that maintains competitive performance with substantially reduced resource requirements, providing a practical solution for deploying large-scale multimodal models.

There are several avenues for future work that warrant investigation. Firstly, ternarization-aware distillation from pre-trained models reduces the student model’s ability to adapt to domain-specific applications. Retraining is still necessary to achieve better performance on specific tasks. Secondly, quantization introduces information loss, which can hinder the model’s ability to capture subtle image-text alignments compared with full-precision models trained from scratch. Thirdly, the current implementation only quantizes weights while leaving activations in FP16 format. Future work could explore activation quantization, such as INT8 and specialized hardware implementations that leverage bit-wise operations for ternary computations, potentially achieving further efficiency gains. Lastly, this work focuses on image-text alignment tasks. The extensions of ternary quantization techniques to other modalities or architectures remain unexplored and present opportunities for future research.

Acknowledgments and Disclosure of Funding

Shao-Qun Zhang is the corresponding author, with email zhangsq@lamda.nju.edu.cn. This research was supported by the Jiangsu Provincial Natural Science Foundation Youth Project (BK20230782).

References

- [1] Hande Alemdar, Vincent Leroy, Adrien Prost Boucle, and Frédéric Pérot. Ternary neural networks for resource-efficient AI applications. In *Proceedings of the 28th International Joint Conference on Neural Networks*, pages 2547–2554, 2017.
- [2] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Advances in Neural Information Processing Systems 32*, pages 7950–7958, 2019.
- [3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pages 1877–1901, 2020.
- [5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.
- [6] Peng Chen, Bohan Zhuang, and Chunhua Shen. FATNN: Fast and accurate ternary neural networks. In *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*, pages 5219–5228, 2021.
- [7] Yifan Chen, Xiaozhen Qiao, Zhe Sun, and Xuelong Li. ComKD-CLIP: Comprehensive knowledge distillation for contrastive language-image pre-training. *arXiv preprint arXiv:2408.04145*, 2024.
- [8] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, pages 4793–4801, 2019.
- [9] Jia Deng, Wei Dong, Richard Socher, Lijia Li, Kai Li, and Li Feifei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 22nd Computer Vision and Pattern Recognition*, pages 248–255, 2009.

- [10] Jacob Devlin, Mingwei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.
- [11] Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. BitDistiller: Unleashing the potential of sub-4-bit LLMs via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 102–116, 2024.
- [12] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *Proceedings of the 8th International Conference on Learning Representations*, pages 1–12, 2020.
- [13] Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10112–10124, 2023.
- [14] Minghao Fu, Hao Yu, Jie Shao, Junjie Zhou, Ke Zhu, and Jianxin Wu. Quantization without tears. In *Proceedings of the 35th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4462–4472, 2025.
- [15] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.
- [16] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *Proceedings of the 4th International Conference on Learning Representations*, 2016.
- [17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems Deep Learning Workshop 2*, 2015.
- [18] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.
- [19] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [20] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

- [21] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 4163–4174, 2020.
- [22] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Feifei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.
- [23] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [24] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-ViT: Accurate and fully quantized low-bit vision transformer. In *Advances in Neural Information Processing Systems 35*, pages 34451–34463, 2022.
- [25] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. In *Proceedings of the 9th International Conference on Learning Representations*, pages 1–15, 2021.
- [26] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. RepQ-ViT: Scale Reparameterization for Post-Training Quantization of Vision Transformers. In *Proceedings of the 19th IEEE/CVF International Conference on Computer Vision*, pages 17227–17236, 2023.
- [27] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021.
- [28] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Weiming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of 7th Annual Conference on Machine Learning and Systems*, pages 87–100, 2024.
- [29] Tsungyi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision*, pages 740–755, 2014.
- [30] Bin Liu, Fengfu Li, Xiaoxing Wang, Bo Zhang, and Junchi Yan. Ternary weight networks. In *Proceedings of 48th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5, 2023.
- [31] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023.

- [32] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *Advances in Neural Information Processing Systems 34*, pages 28092–28103, 2021.
- [33] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020.
- [34] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *Proceedings of the 6th International Conference on Learning Representations*, pages 1–9, 2018.
- [35] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *Proceedings of the 6th International Conference on Learning Representations*, pages 1–10, 2018.
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.
- [37] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. In *Proceedings of the 14th European Conference on Computer Vision*, pages 525–542, 2016.
- [38] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–13, 2015.
- [39] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.
- [40] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.
- [41] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.
- [42] Vivienne Sze, Yuhsin Chen, Tienju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

- [43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [44] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 210–218, 2018.
- [45] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.
- [46] Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi Chen, Xinggang Wang, Hongyang Chao, and Han Hu. TinyCLIP: CLIP distillation via affinity mimicking and weight inheritance. In *Proceedings of the 19th IEEE/CVF International Conference on Computer Vision*, pages 21970–21980, 2023.
- [47] Chuanguang Yang, Zhulin An, Libo Huang, Junyu Bi, Xinqiang Yu, Han Yang, Boyu Diao, and Yongjun Xu. CLIP-KD: An empirical study of CLIP model distillation. In *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18773–18782, 2024.
- [48] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets. In *Proceedings of the 7th International Conference on Learning Representations*, pages 1–17, 2019.
- [49] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.
- [50] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8BERT: Quantized 8bit BERT. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition*, pages 36–39, 2019.
- [51] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- [52] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. TernaryBERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing*, pages 509–521, 2020.
- [53] Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. Trained ternary quantization. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–10, 2017.

- [54] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*, 2023.

A. Training Hyperparameters

In Table 6, we exhibit hyperparameters used to train our TernaryCLIP model, while the values between square brackets are the to-be-selected hyperparameters and the values with the text bold font are the selected hyperparameters after the tuning procedure.

Hardware	8 * NVIDIA RTX 6000 Ada
Model	Teacher: [ViT-L/14] Student: [ViT-B/16]
Weight	[OpenAI_400M_ep32, Laion400M_ep32]
Quantization	weight_quant: [ternary , int3, int4] activation_quant: [int8, float16]
Ternarization	β : [1, 2 , 3], ϵ : [1e-6]
Precision	[amp, amp_bf16, bp16, fp32]
Data Load	num_workers: [4, 8, 16 , 32]
Epoch	[32, 64]
Learning Rate	[1e-4, 5e-4, 1e-3]
Warmup	[1000, 5000, 10000 , 100000]
Batch Size	[128×8, 256×8, 384 ×8, 512×8]
Optimizer	adamw(0.9 , [0.98, 0.998, 0.999], ϵ : [1e-6])
Weight Decay	[0.01, 0.05, 0.1 , 0.2]
$\lambda_{\text{crd}}, \tau_{\text{crd}}$	[0.5, 1 , 2], [0.5, 1 , 2]
λ_{icl}	[0.5, 1 , 2]
λ_{fd}	[1000, 2000 , 4000]
Augment Config	[None , Scale, Scale+Color_Jitter+Gray_Scale]

Table 6: Training hyperparameters of TernaryCLIP: the value with bold font is the recommended hyperparameters after tuning. For example, $\lambda_{\text{crd}} = 1.0$ and temperature $\tau_{\text{crd}} = 1.0$.

B. Details of Training TernaryCLIP

In Figure 7, 8, 9, and 10, we show the distinct loss curves of three models on the number of samples seen, as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{crd}} + \mathcal{L}_{\text{icl}} + \mathcal{L}_{\text{df}}$. The first model is full-precision CLIP-KD, the second model is TernaryCLIP_Q-FFN with ternarized FFN, and the third model is TernaryCLIP_Q-ALL with ternarized MHA+FFN. The first loss curve is task loss. The second loss curve is contrastive relational distillation (CRD) loss. The third loss curve is interactive contrastive learning (ICL) loss. The last loss curve is feature distillation (FD) loss. The loss curves are plotted against the number of samples seen during training.

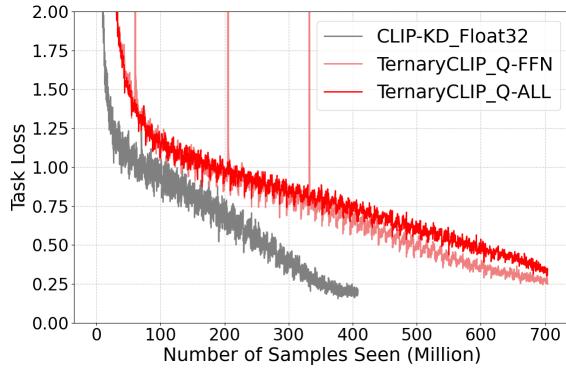


Figure 7: Task losses of full-precision CLIP-KD, TernaryCLIP_Q-FFN and TernaryCLIP_Q-ALL.

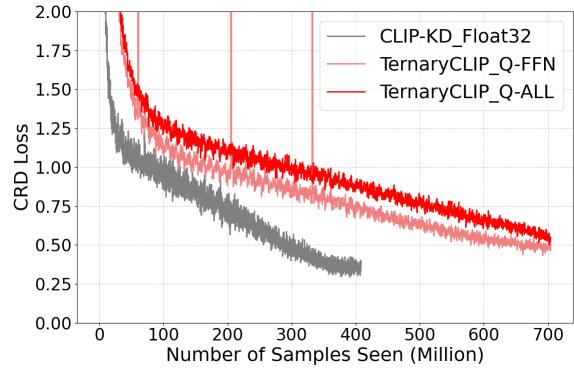


Figure 8: CRD losses of full-precision CLIP-KD, TernaryCLIP_Q-FFN and TernaryCLIP_Q-ALL.

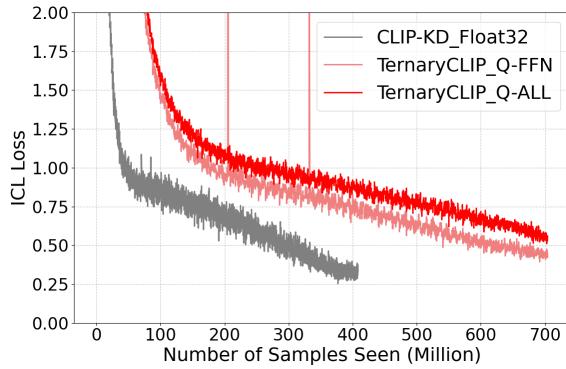


Figure 9: ICL losses of full-precision CLIP-KD, TernaryCLIP_Q-FFN and TernaryCLIP_Q-ALL.

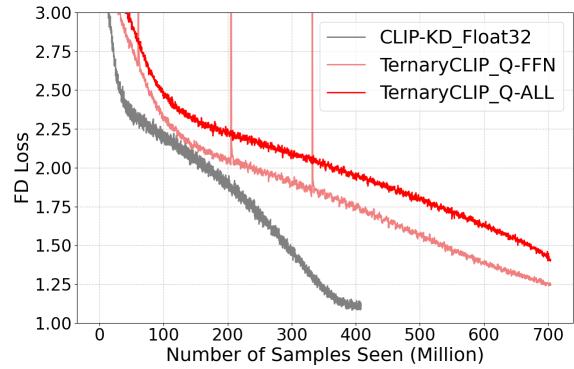


Figure 10: FD losses of full-precision CLIP-KD, TernaryCLIP_Q-FFN and TernaryCLIP_Q-ALL.

C. Details of Zero-Shot Image Classification

In Table 7, TnCLIP represents the abbreviation of TernaryCLIP. In Figure 11 and Table 7, TernaryCLIP_Q-ALL with 99% ternary weights obtains a 2.71% performance reduction compared with 32-bit CLIP-KD, and TernaryCLIP_Q-FFN with 72% ternary weights gets a 1.83% performance degradation. Notably, TernaryCLIP_Q-ALL achieves a 5.29% performance improvement compared with RepQ+QwT of 4-bit quantized weights and 32-bit compensation weights, and TernaryCLIP_Q-FFN gains a 6.17% performance improvement.

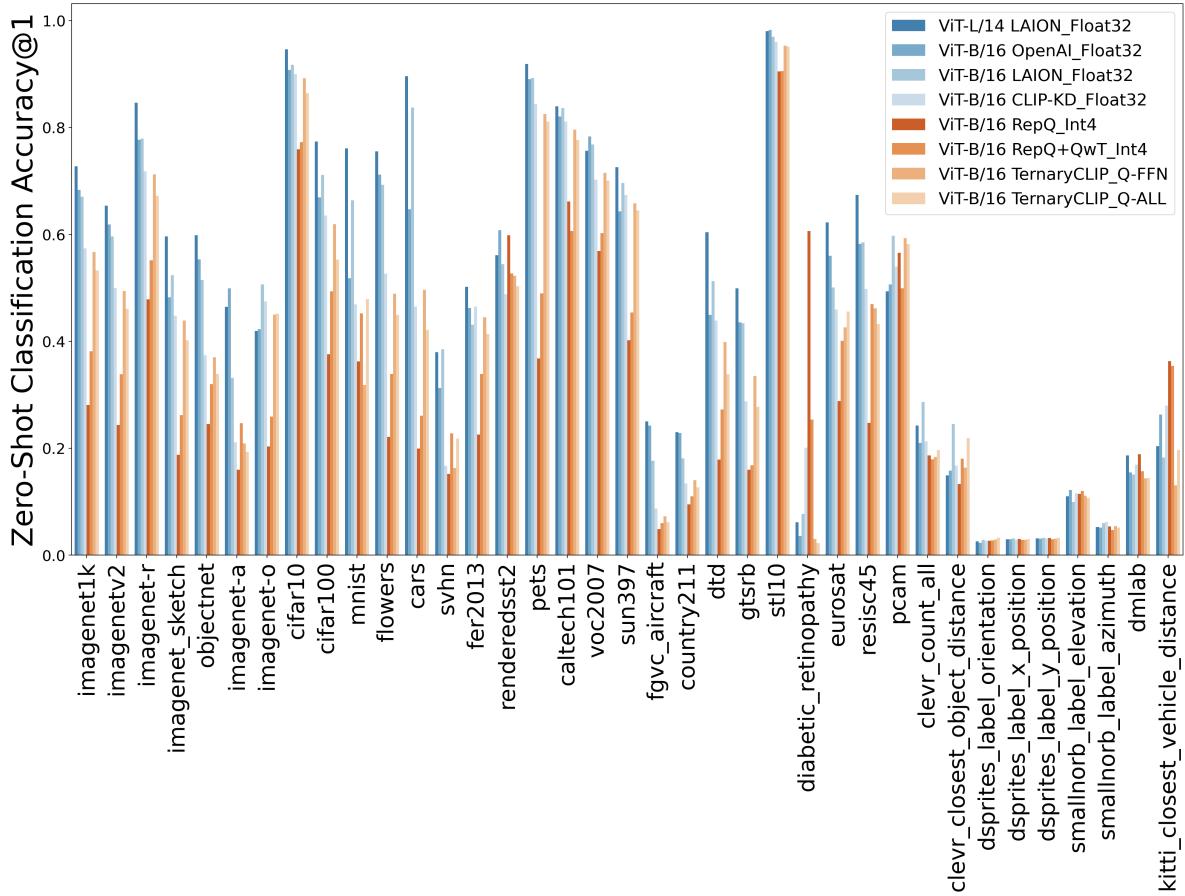


Figure 11: Zero-shot image classification performance: Accuracy@1 across 37 datasets.

Datasets	ViT-L/14 LAION _Float32	ViT-B/16 OpenAI _Float32	ViT-B/16 LAION _Float32	ViT-B/16 CLIP-KD _Float32	ViT-B/16 RepQ _Int4	ViT-B/16 RepQ+QwT _Int4	ViT-B/16 TnCLIP _Q-FFN	ViT-B/16 TnCLIP _Q-ALL
<i>Natural Datasets</i>								
caltech101	83.99%	82.10%	83.63%	81.15%	66.16%	60.64%	79.61%	77.70%
cars	89.64%	64.73%	83.77%	46.50%	19.95%	26.07%	49.67%	42.20%
cifar10	94.63%	90.77%	91.73%	89.99%	75.96%	77.26%	89.24%	86.45%
cifar100	77.39%	66.94%	71.15%	63.57%	37.62%	49.40%	61.96%	55.28%
country211	23.04%	22.87%	18.12%	13.41%	9.52%	11.03%	14.04%	12.68%
dtd	60.43%	44.95%	51.28%	43.88%	17.87%	27.29%	39.89%	33.83%
fer2013	50.22%	46.22%	43.13%	46.53%	22.58%	33.91%	44.52%	41.40%
fgvc_aircraft	25.02%	24.24%	17.64%	8.73%	4.89%	5.97%	7.29%	6.15%
flowers	75.56%	71.18%	69.28%	52.66%	22.13%	33.92%	48.92%	44.97%
gtsrb	49.92%	43.56%	43.42%	28.79%	15.98%	16.85%	33.49%	27.75%
imagenet-a	46.49%	49.92%	33.19%	21.12%	15.97%	24.69%	20.91%	19.36%
imagenet-o	41.95%	42.30%	50.65%	47.45%	20.35%	25.95%	45.00%	45.20%
imagenet-r	84.68%	77.71%	77.93%	71.83%	47.84%	55.20%	71.28%	67.22%
imagenet1k	72.77%	68.36%	67.07%	57.44%	28.10%	38.17%	56.76%	53.28%
imagenetv2	65.41%	61.90%	59.65%	50.01%	24.38%	33.85%	49.43%	46.10%
objectnet	59.86%	55.33%	51.49%	37.41%	24.56%	32.00%	37.02%	33.89%
pets	91.91%	89.04%	89.26%	84.44%	36.82%	49.01%	82.56%	81.14%
stl10	98.05%	98.25%	96.99%	96.05%	90.53%	90.56%	95.29%	95.16%
sun397	72.59%	64.34%	69.61%	67.41%	40.23%	45.43%	65.84%	64.48%
svhn	37.97%	31.27%	38.53%	16.72%	15.19%	22.80%	16.33%	21.79%
voc2007	75.64%	78.32%	76.85%	70.23%	56.92%	60.26%	71.51%	70.10%
<i>Specialized Datasets</i>								
diabetic_retinopathy	6.19%	3.57%	7.75%	20.09%	60.67%	25.36%	3.03%	2.27%
eurosat	62.24%	56.00%	50.07%	45.96%	28.85%	40.09%	42.59%	45.59%
imagenet_sketch	59.65%	48.24%	52.37%	44.79%	18.81%	26.22%	43.91%	40.23%
mnist	76.10%	51.85%	66.39%	46.90%	36.27%	45.25%	31.86%	47.93%
pcam	49.38%	50.67%	59.73%	53.98%	56.58%	49.94%	59.30%	58.18%
renderedsst2	56.12%	60.79%	54.48%	48.82%	59.86%	52.72%	52.28%	50.30%
resisc45	67.43%	58.27%	58.54%	49.83%	24.73%	46.95%	46.21%	43.24%
<i>Structured Datasets</i>								
clevr_closest_object_distance	14.91%	15.83%	24.51%	16.75%	13.29%	18.07%	16.40%	21.90%
clevr_count_all	24.25%	21.03%	28.65%	21.27%	18.65%	17.93%	18.37%	19.65%
dmlab	18.66%	15.50%	15.10%	16.92%	18.88%	15.70%	14.39%	14.40%
dsprites_label_orientation	2.61%	2.34%	2.87%	2.71%	2.71%	2.77%	2.92%	3.28%
dsprites_label_x-position	2.99%	3.00%	3.15%	2.93%	3.05%	2.90%	2.86%	3.04%
dsprites_label_y-position	3.16%	3.11%	3.24%	3.16%	3.20%	3.01%	3.07%	3.28%
kitti_closest_vehicle_distance	20.39%	26.30%	18.28%	27.99%	36.29%	35.44%	13.08%	19.69%
smallnorb_label_azimuth	5.25%	5.18%	6.02%	6.20%	5.39%	4.71%	5.46%	5.16%
smallnorb_label_elevation	11.00%	12.21%	9.96%	11.59%	11.49%	12.00%	11.11%	10.71%
<i>Summary</i>								
Average (natural)	65.58%	60.68%	61.16%	52.16%	33.03%	39.06%	51.46%	48.86%
Average (specialized)	53.87%	47.06%	49.90%	44.34%	40.82%	40.93%	39.88%	41.11%
Average (structured)	11.47%	11.61%	12.42%	12.17%	12.55%	12.50%	9.74%	11.23%
Average (All)	50.20%	46.17%	47.18%	40.95%	29.52%	32.95%	39.12%	38.24%
Perf vs. ViT-L/14 LAION	0.00%	-4.03%	-3.02%	-9.25%	-20.68%	-17.25%	-11.08%	-11.96%
Perf vs. ViT-B/16 OpenAI	4.03%	0.00%	1.01%	-5.22%	-16.65%	-13.22%	-7.05%	-7.93%
Perf vs. ViT-B/16 LAION	3.02%	-1.01%	0.00%	-6.23%	-17.66%	-14.23%	-8.06%	-8.94%
Perf vs. ViT-B/16 CLIP-KD	9.25%	5.22%	6.23%	0.00%	-11.43%	-8.00%	-1.83%	-2.71%
Perf vs. ViT-B/16 RepQ	20.68%	16.65%	17.66%	11.43%	0.00%	3.43%	9.60%	8.72%
Perf vs. ViT-B/16 RepQ+QwT	17.25%	13.22%	14.23%	8.00%	-3.43%	0.00%	6.17%	5.29%

Table 7: Zero-shot image classification performance: Accuracy@1 on 37 datasets covering 3 distinct types.

D. Details of Inference Latency

In Table 8, we present a comprehensive analysis of CLIP model inference latency across various precision formats with the corresponding practical bit-per-weight (BPW) implementations by the GGML library, indicating the decomposition of total latency into model loading, image loading, and model forwarding components, with all benchmarks conducted on Apple M4 Pro hardware over 1,000 test rounds.

Models	Precision	BPW ↓	Storage(MB) ↓	Model Load(ms) ↓	Image Load(ms) ↓	Model Forward(ms) ↓	Total Latency(ms) ↓
ViT-L/14	Float32	32	1630.90	382.22 ± 22.58	5.32 ± 1.65	498.60 ± 85.66	886.15 ± 90.65
	Float16	16	817.28	199.42 ± 9.50	5.35 ± 0.24	506.10 ± 75.54	710.88 ± 76.38
ViT-B/16	Float32	32	571.60	153.02 ± 7.41	5.35 ± 0.22	83.02 ± 55.47	241.40 ± 56.47
	Float16	16	287.73	90.15 ± 4.27	5.42 ± 0.63	79.04 ± 50.26	174.62 ± 50.52
TernaryCLIP.Q_FFN ViT-B/16	Float32	32	571.60	154.55 ± 21.00	5.79 ± 0.77	90.03 ± 59.39	250.38 ± 67.65
	Float16	16	287.73	90.52 ± 4.87	5.84 ± 0.72	86.62 ± 53.00	182.99 ± 53.57
	Q4.0	4	140.93	57.32 ± 4.35	5.83 ± 0.44	74.02 ± 49.97	137.18 ± 50.52
	Q4.1	4	147.31	58.53 ± 2.86	5.82 ± 0.38	70.40 ± 48.54	134.75 ± 48.68
	TQ1_0	1.6875	105	49.09 ± 3.47	5.81 ± 0.33	71.14 ± 50.89	126.05 ± 51.05
TernaryCLIP.Q_ALL ViT-B/16	Float32	32	571.60	149.82 ± 11.73	5.63 ± 0.45	87.83 ± 56.36	243.28 ± 57.14
	Float16	16	287.73	89.12 ± 3.77	5.77 ± 0.32	79.70 ± 50.35	174.60 ± 50.40
	Q4.0	4	84.87	44.16 ± 1.79	5.75 ± 0.24	71.30 ± 49.68	121.22 ± 49.79
	Q4.1	4	93.69	46.10 ± 1.87	5.75 ± 0.29	68.46 ± 46.07	120.32 ± 46.08
	TQ1_0	1.6875	35.25	33.05 ± 1.70	5.74 ± 0.19	67.96 ± 45.58	106.75 ± 45.68

Table 8: CLIP model inference latency overview on different precisions and bpw (bits per weight). The CPU hardware is Apple M4 Pro. Total latency is combined with model loading, image loading, and model forwarding. Latency = average value ± three times standard deviation under 1,000 rounds of benchmarks.

E. Ablation Study

In Figure 12 and Figure 13, we conducted an ablation study of data augmentation and int8 activation quantization. Based on the performance gap of loss curves, we determine training TernaryCLIP without any data augmentation and activation quantization.

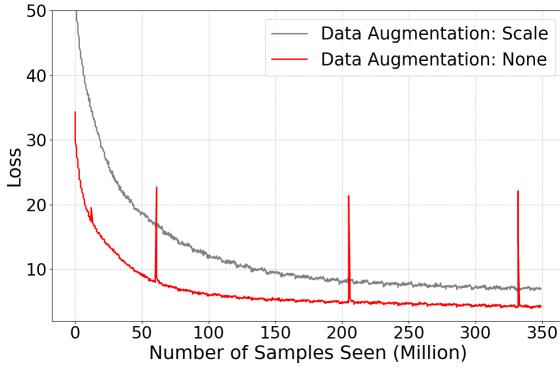


Figure 12: Ablation study of data augmentation on TernaryCLIP. Data augmentation makes loss convergence much slower than the baseline.

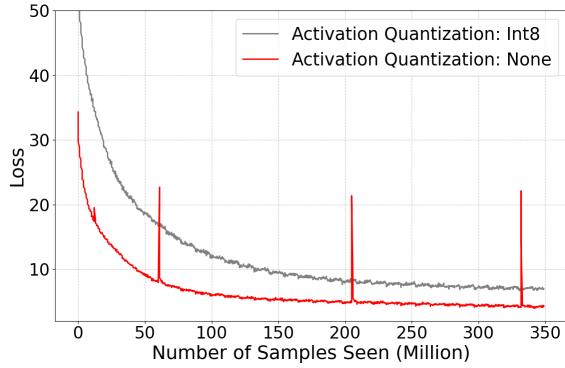


Figure 13: Ablation study of int8 or float16 activation quantization on TernaryCLIP. The int8 activation quantization makes loss convergence much slower than the baseline.

In Figure 14 and Figure 15, we conducted an ablation study on the effect of self-distillation and distillation with different teacher models. The performance degradation of self-distillation is only 1.35% compared with distillation, which is acceptable for the sake of training efficiency and simplicity. The results show that self-distillation achieves competitive performance as distillation under the same training configurations, which indicates that ternarization-aware distillation is not limited to the larger teacher model.

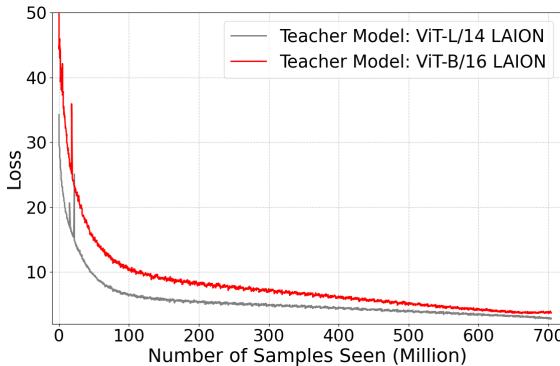


Figure 14: Ablation study of self-distillation and distillation on TernaryCLIP training. Self-distillation maintains similar convergence characteristics.

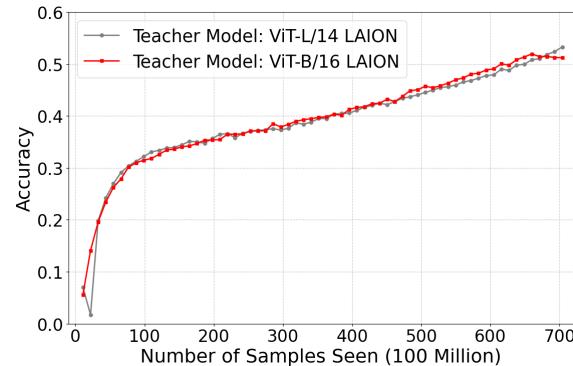


Figure 15: Ablation study of self-distillation and distillation on TernaryCLIP accuracy. Self-distillation achieves only 1.35% performance degradation.

F. Ternary Weight Distribution

In Figure 16, Figure 17, and Figure 18, we illustrate the distribution of ternary weight for TernaryCLIP Q-FFN and Q-ALL models. It is obvious that ternary weights maintain a good sparsity in addition to other benefits of quantization that we have discussed during experiments.

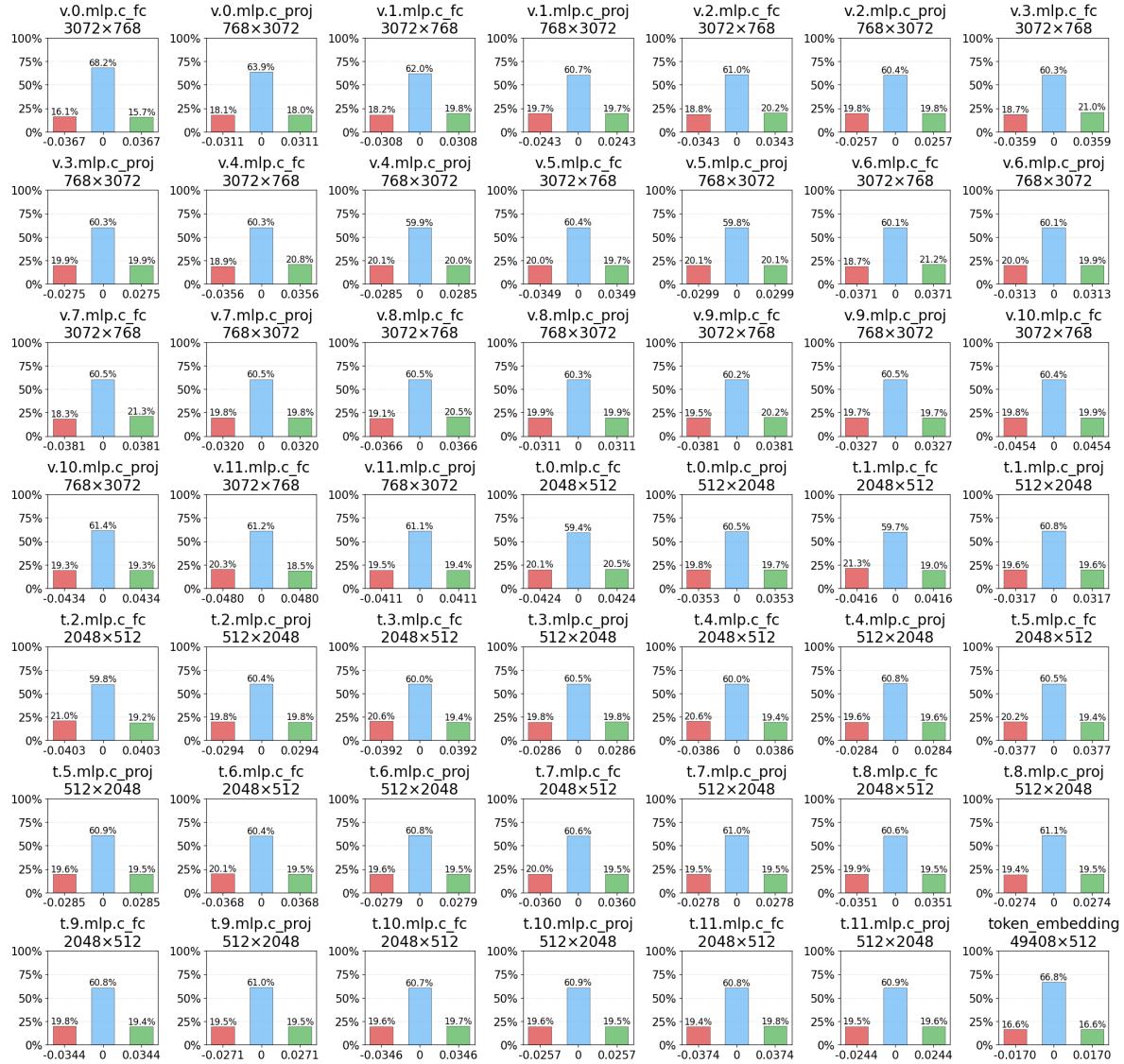


Figure 16: Ternary weight distribution of TernaryCLIP_Q-FFN.



Figure 17: Ternary weight distribution of TernaryCLIP_Q-ALL (Part1).

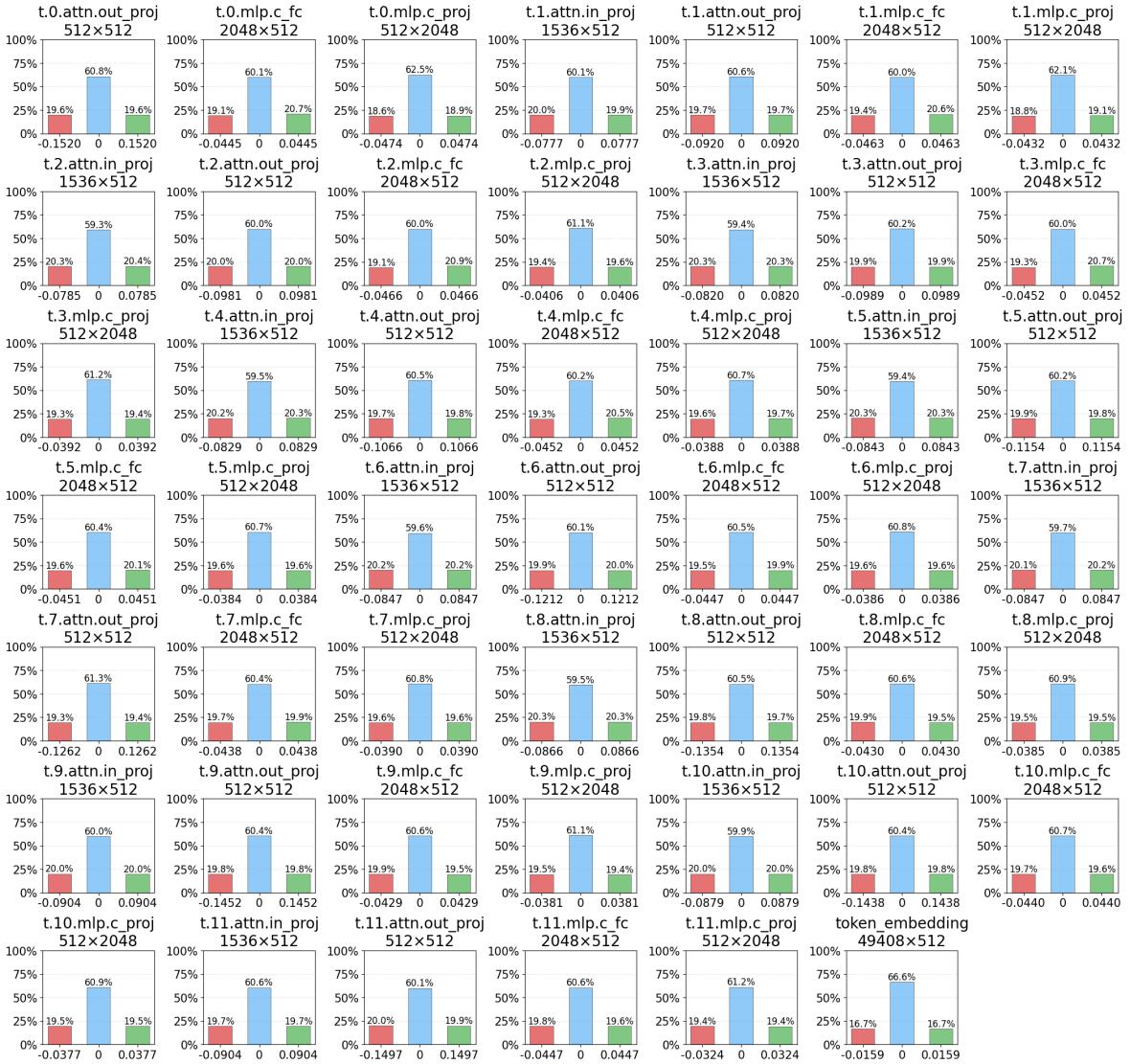


Figure 18: Ternary weight distribution of TernaryCLIP_Q-ALL (Part2).