Decompose, Mix, Adapt: A Unified Framework for Parameter-Efficient Neural Network Recombination and Compression

Supplementary Material

1. CRISP Algorithms

CRISP consists of three core stages: weight reparameterization via the forward pass (Alg. 1), neural mimicry retrofitting (Alg. 2), and task adaptation via PEFT (Alg. 3). The compression stage differs between ViT and LLaMA architectures and is described separately below. The complete joint MC+PEFT pipeline is presented in Alg. 6.

Algorithm 1 describes the forward pass using the coefficient-gated transformation $\mathcal{T}_{\text{CRISP}}(B'_i, A'^{rs}_i) = B'_i(\sigma(A'^{rs}_i) \odot A'^{rs}_i)$ introduced in Eq. 4 of the main paper. The retrofitting process in Algorithm 2 implements the initial neural mimicry stage (Sec. 3.2), which decomposes pretrained weights into basis-mixer pairs through smooth-L1 reconstruction loss without requiring any dataset samples. Algorithm 3 demonstrates CRISP's PEFT capability, where only the lightweight mixer matrices $\{A'^{rs}_i\}$ are updated while basis matrices $\{B'_i\}$ remain frozen, enabling task adaptation with fewer than 200 trainable parameters per layer in some experiments (Tab. 1, main paper).

Algorithm 1 CRISP Forward Pass

Require: Input $\mathbf{x} \in \mathbb{R}^{n \times d_{\text{in}}}$, Basis $B'_i \in \mathbb{R}^{u \times r}$, Mixer $A'^{rs}_i \in \mathbb{R}^{r \times s}$, bias $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$
Ensure: Output $\mathbf{y} \in \mathbb{R}^{n \times d_{\text{out}}}$

$\tilde{A}'^{rs}_i \leftarrow \sigma(A'^{rs}_i) \odot A'^{rs}_i$ where $\sigma(\cdot)$ is sigmoid
$W_i \leftarrow \text{reshape}(B'_i \tilde{A}'^{rs}_i, (d_{\text{out}}, d_{\text{in}}))$ where $u = \frac{d_{\text{in}} \cdot d_{\text{out}}}{s}$
$\mathbf{y} \leftarrow \mathbf{x}W_i^T + \mathbf{b}$
return $\mathbf{y}$

Algorithm 2 Neural Mimicry Initialization

Require: Pretrained weights $\{W^i_p\}_{i=1}^N$, layer groups $\mathcal{G}$, hyperparameters r, s
Ensure: Shared bases $\{B'_i\}$, per-layer mixers $\{A'^{rs}_i\}$

for each group $g \in \mathcal{G}$ **do**
  Initialize shared basis $B'_g \sim \mathcal{N}(0, 0.01)$ of size $u \times r$
  for each layer $i \in g$ **do**
    Initialize mixer $A'^{rs}_i \sim \mathcal{N}(0, 0.01)$ of size $r \times s$
  end for
end for
while not converged **do**
  $\mathcal{L}_{\text{mimicry}} \leftarrow \sum_{i=1}^N \ell_{\text{smL1}}(\mathcal{T}_{\text{CRISP}}(B'_i, A'^{rs}_i) - W^i_p)$
  Update $\{B'_i\}, \{A'^{rs}_i\}$ via gradient descent on $\mathcal{L}_{\text{mimicry}}$
end while
return $\{B'_i\}, \{A'^{rs}_i\}$

Algorithm 3 CRISP PEFT Adaptation

Require: Downstream dataset $\mathcal{D}$, compressed model with frozen bases $\{B'_i\}$, trainable mixers $\{A'^{rs}_i\}$, learning rate η
Ensure: Task-adapted model

for $(\mathbf{x}, y) \in \mathcal{D}$ **do**
  $\hat{y} \leftarrow \text{Forward}(\mathbf{x}; \{B'_i\}, \{A'^{rs}_i\})$ ▷ Alg. 1
  $\mathcal{L}_{\text{task}} \leftarrow \text{CrossEntropy}(\hat{y}, y)$
  $\{A'^{rs}_i\} \leftarrow \{A'^{rs}_i\} - \eta \nabla_{\{A'^{rs}_i\}} \mathcal{L}_{\text{task}}$ ▷ Freeze $\{B'_i\}$
end for
return adapted $\{A'^{rs}_i\}$

1.1. ViT Compression

For ViT models, compression is performed via distillation using a full-parameter CRISP teacher model trained on only 2% of ImageNet-1K [1]. Algorithm 4 initializes the student model using the top-r eigenvectors of the teacher's basis matrices and optimizes a weighted combination of KL divergence on output logits and per-layer MSE feature matching (Tab. 2, main paper; Supp. Tabs. 3 and 4).

1.2. LLaMA Compression

For LLaMA models, compression operates differently from the ViT setting. Algorithm 5 first performs data-free basis reduction by computing importance scores from template norms and coefficient sparsity, clustering basis vectors via importance-weighted k-means, and merging clusters with variance-aware rescaling. A short calibration stage (Stage 2) refines the compressed model using a weighted combination of weight reconstruction and language modeling loss, yielding further gains (Tab. 6).

1.3. Joint MC+PEFT Pipeline

Algorithm 6 presents the complete pipeline for simultaneous compression and task adaptation. Compression (ViT: Alg. 4; LLaMA: Alg. 5) is applied first, after which basis matrices are frozen and only mixer matrices are updated for downstream tasks via Alg. 3. This unified pipeline achieves state-of-the-art on both MC (Tab. 2, main paper; Tab. 6) and PEFT (Tab. 1, main paper; Tab. 5) without requiring separate optimization procedures.

Algorithm 4 CRISP-ViT Compression via Distillation

Require: Teacher model $\mathcal{M}_{\text{teacher}}$ (full CRISP from Alg. 2), target compression $r_{\text{target}}, s_{\text{target}}$, distillation dataset $\mathcal{D}_{\text{dist}}$ (2% ImageNet), loss weights $\lambda_{\text{KL}}, \lambda_{\text{feat}}$
Ensure: Compressed student model $\mathcal{M}_{\text{student}}$
Initialize $\{B'_{\text{student}}\}, \{A'^{rs}_{\text{student}}\}$ with top r_{target} eigenvectors from $\mathcal{M}_{\text{teacher}}$
for $(\mathbf{x}, y) \in \mathcal{D}_{\text{dist}}$ **do**
  $\hat{y}_{\text{teacher}} \leftarrow \mathcal{M}_{\text{teacher}}(\mathbf{x})$
  $\hat{y}_{\text{student}} \leftarrow \mathcal{M}_{\text{student}}(\mathbf{x})$
  $\mathcal{L}_{\text{KL}} \leftarrow \text{KL}(\hat{y}_{\text{student}} \| \hat{y}_{\text{teacher}})$ ▷ Output logit distillation
  $\mathcal{L}_{\text{feat}} \leftarrow \sum_{\ell} \text{MSE}(f_{\ell}^{\text{student}}, f_{\ell}^{\text{teacher}})$ ▷ Per-layer feature alignment
  $\mathcal{L} \leftarrow \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}$
  Update $\{B'_{\text{student}}\}, \{A'^{rs}_{\text{student}}\}$ via gradient descent on $\mathcal{L}$
end for
return $\mathcal{M}_{\text{student}}$

2. Datasets

PEFT Evaluation (ViT). Tab. 1 details the 19 tasks in the VTAB-1K [16] benchmark used to evaluate PEFT on ViT-S/16. Tasks are grouped into three categories: Natural (7 tasks), Specialized (4 tasks, including medical and satellite imagery), and Structured (8 tasks requiring geometric and relational reasoning). Each task provides 1K training samples, making it well-suited for evaluating sample efficiency and cross-domain generalization.

Compression Evaluation (ViT). Tab. 2 details the six fine-grained classification benchmarks used for ViT-B/16 compression evaluation, corresponding to the results in Tab. 2 of the main paper and Tabs. 3 and 4. These datasets span a wider range of training budgets than VTAB-1K, allowing us to assess compression quality under less constrained finetuning conditions.

3. Ablation Studies

We conduct ablation studies to validate key design choices in CRISP's architecture, focusing on three critical components: constraint placement, initialization strategies, and reconstruction loss functions during neural mimicry.

Regularization via Coefficient Constraints. Fig. 1 compares three strategies for where to apply the regularization constraint in the CRISP transformation: PRE applies the constraint to the mixer before combining $\sigma(A'^{rs}) \odot A'^{rs}$), POST applies it after reconstruction ($\phi(B'A'^{rs})$), and TEMP applies it to the basis matrices ($B'\phi(A'^{rs})$). Our formulation (PRE with SiLU-style gating) consistently outperforms alternatives across all benchmarks, providing implicit regularization on the mixer coefficients without constraining the final weight space. POST and TEMP configurations can slightly hurt performance as they constrain the output weight space directly, which acts as a hard constraint on the layer weights as discussed in Sec. 3.1 of the main paper. Notably, ReLU performs catastrophically

Algorithm 5 CRISP-Llama Compression Pipeline

Require: CRISP model with template banks $\{B_g^{(p)}\}$, coefficients $\{A_i^{(p)}\}$, per-projection compression rates $\{\rho_p\}$, calibration data $\mathcal{D}_{\text{cal}}$ (optional)
Ensure: Compressed model with reduced ranks $r'_p = \lfloor r(1-\rho_p) \rfloor$
Stage 1: Data-Free Basis Reduction
for each group g, projection $p \in$ {up, gate, down, q, k, v, o} **do**
  $w \leftarrow \text{IMPORTANCE}(B_g^{(p)}, \{A_i^{(p)}\}, \mathcal{D}_{\text{cal}})$ ▷ Eq. 1 + optional variance
  $\pi \leftarrow \text{CLUSTER}(B_g^{(p)}, \{A_i^{(p)}\}, w, r'_p)$ ▷ KMeans + random projection
  $B_g^{(p)} \leftarrow \text{MERGE}(B_g^{(p)}, \pi, w)$ ▷ Weighted avg. + rescaling
  $\{A_i^{(p)}\} \leftarrow \text{AGGREGATE}(\{A_i^{(p)}\}, \pi)$ ▷ Sum per cluster
end for
Stage 1b: Coefficient Re-solve ▷ Least-squares: $A_i^{(p)} \leftarrow W_{\text{teacher}}^{(p)}(B_g^{(p)})^{\dagger}$
Stage 2: Activation Calibration
for $e = 1$ to E_{calib} **do**
  $\mathcal{L} \leftarrow w_1 \sum_{i,p} \|W_i^{(p)} - W_{\text{teacher}}^{(p)}\|_F^2 + w_2\mathcal{L}_{\text{LM}}$
  Update $\{B_g^{(p)}\}, \{A_i^{(p)}\}$ via gradient descent
end for
return Compressed model

Sub-procedures:

function IMPORTANCE($B, \{A_i\}, \mathcal{D}$)
  $w \leftarrow \|B\|_2 + \lambda \sum_i \|A_i\|_1$ ▷ Per Eq. 1; boost by σ^2 if $\mathcal{D}$ provided
  return w
end function
function CLUSTER($B, \{A_i\}, w, r'$)
  $\Phi \leftarrow [\text{RandomProj}(B); \text{concat}(A_i)]$ weighted by w
  return KMeans(Φ, r')
end function
function MERGE(B, π, w)
  return Importance-weighted average with variance rescaling
end function
function AGGREGATE($\{A_i\}, \pi$)
  return Sum coefficients per cluster
end function

Algorithm 6 CRISP Joint MC+PEFT

Require: Pretrained weights $\{W_p^i\}$, target compression rate, downstream task $\mathcal{D}$

Ensure: Compressed and task-adapted model

1: $\{B'\}, \{A'^{rs}\} \leftarrow \text{NeuralMimicry}(\{W_p^i\}, r_{\text{full}}, s_{\text{full}})$ ▷ Alg. 2
2: $\mathcal{M}_{\text{student}} \leftarrow \text{Compress}(\mathcal{M}_{\text{teacher}}, r_{\text{target}}, s_{\text{target}})$ ▷ Alg. 4
3: Freeze $\{B'_{\text{student}}\}$
4: $\{A'^{rs}_{\text{adapted}}\} \leftarrow \text{PEFT}(\mathcal{D}, \{B'_{\text{student}}\}, \{A'^{rs}_{\text{student}}\})$ ▷ Alg. 3
5: **return** $\mathcal{M}_{\text{student}}$ with $\{B'_{\text{student}}\}, \{A'^{rs}_{\text{adapted}}\}$

Table 1. Details of VTAB-1K [16] Benchmark used for PEFT on ViT-S/16 [2].

Dataset	#Cat	#Train	#Val	#Test
CIFAR100	100	800/1000	200	10000
Caltech101	102	6084	–	–
DTD	47	1880	–	–
Flower102	102	6149	–	–
Pets	37	3669	–	–
SVHN	10	26032	–	–
Sun397	397	21750	–	–
Camelyon	2	800/1000	200	32768
EuroSAT	10	5400	–	–
Resisc45	45	6300	–	–
Retinopathy	5	42670	–	–
Clevr-Count	8	800/1000	200	15000
Clevr-Dist	6	15000	–	–
DMLab	6	22735	–	–
KITTI-Dist	4	711	–	–
dSpr-Loc	16	73728	–	–
dSpr-Ori	16	73728	–	–
sNORB-Azim	18	12150	–	–
sNORB-Ele	9	12150	–	–

Table 2. Details on the Benchmarking Datasets used for fine-tuning Compressed ViT-B/16 [2]

Dataset	Classes	#Sample
Oxford Flowers [9]	102	6553
FGVC Aircrafts [8]	55	10001
MIT Scenes [11]	67	15614
CIFAR100 [6]	100	60000
CIFAR10 [6]	10	60000
CUBs (Birds) [14]	200	11789

across all placements due to excessive sparsification of the mixer matrices, consistent with the ablation results in Tab. 4 of the main paper. GELU shows competitive performance to SiLU, but SiLU's smooth gating better balances expressivity and regularization without introducing additional hyperparameters.

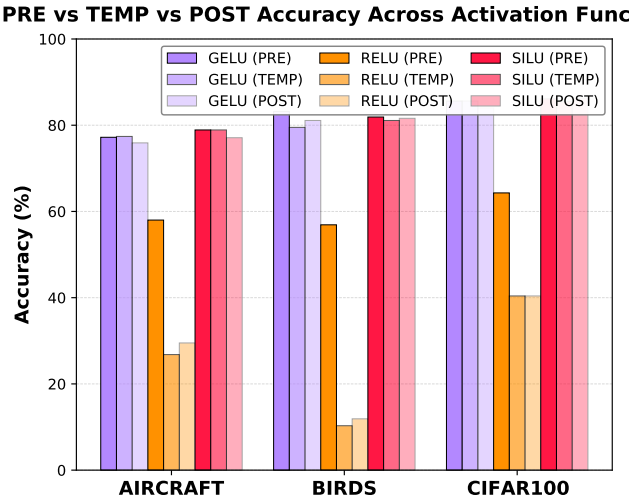

Figure 1. Impact of regularization constraint placement across PRE, POST, and TEMP configurations. PRE (our method) achieves the most consistent performance, while ReLU causes severe degradation due to weight sparsification.

Initialization methods. Fig. 3 evaluates four initialization strategies for the mixer matrices A'^{rs} during Algorithm 2: uniform, Kaiming, Xavier, and orthogonal. Results show remarkable robustness across initialization schemes, with all methods achieving within 1% of each other on most tasks. This insensitivity to initialization validates that the neural mimicry objective effectively guides the learning process regardless of starting point. The slight advantage of Kaiming and orthogonal initializations on certain tasks motivated our choice of orthogonal initialization as mentioned in the paper, but practitioners can confidently use simpler schemes without significant performance degradation.

Loss functions for neural mimicry. Fig. 2 compares reconstruction losses in Algorithm 2: Huber, smooth-L1, MSE, and L1. All robust losses (Huber, smooth-L1) perform comparably, with smooth-L1 showing marginal advantages on fine-grained tasks like Aircraft. The consistent performance across loss functions suggests that the choice of reconstruction objective is less critical than the overall factorization framework, though smooth-L1's robustness to outliers during weight decomposition motivated its adoption in our implementation. Notably, L1 loss shows competitive or superior performance on some tasks, indicating potential for further exploration of sparsity-inducing objectives during retrofitting.

Compression ablations. Tab. 3 evaluates design choices for compressing ViT-B/16 by 50% across six benchmarks (see Tab. 2). Neural mimicry alone (Eq. 5 on main paper) significantly underperforms using distillation, demonstrat-

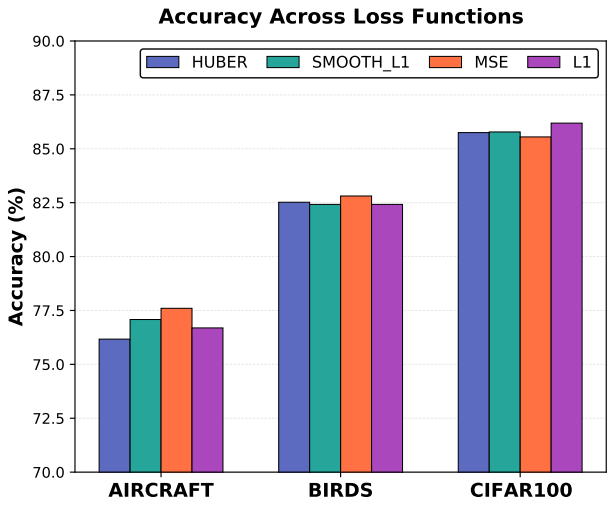

Figure 2. **Effect of reconstruction loss functions during neural mimicry.** We compare four loss functions (Huber, Smooth-L1, MSE, L1) used in the neural mimicry stage (Equation 5 of main paper) for retrofitting pretrained weights into CRISP's basis-mixer decomposition.

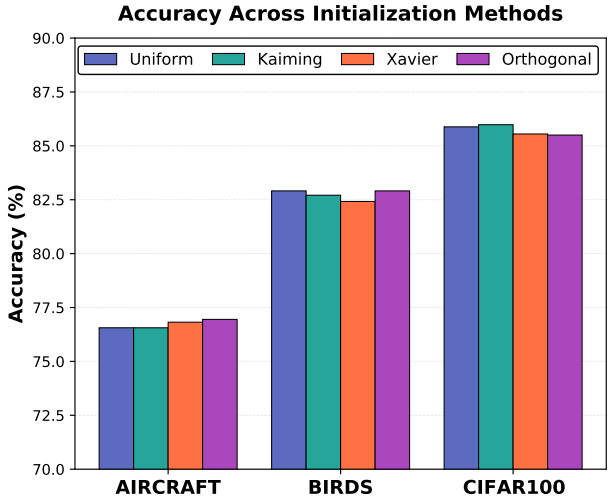

Figure 3. **Robustness to initialization methods.** We evaluate four standard initialization schemes (Uniform, Kaiming, Xavier, Orthogonal) for the mixer matrices A'^{rs} during both neural mimicry retrofitting and subsequent task adaptation.

ing that weight-space reconstruction without data is insufficient for aggressive compression. In contrast, distillation boosts accuracy by 31%, validating our two-stage approach. We also show that using SVD initialization is important (replacing it with random orthogonal initialization) as it provides a boost of almost 4%, confirming that eigenvector-based warm-starting provides a stronger initialization for the compressed parameter space.

Additional ViT Compression Results Tab. 4 reports performance on 75% compression to supplement our 50% compression results from our main paper. We find that CRISP outperforms other PR methods like RECAST [13] by 10% on MC. This gap is increased when combined with PEFT methods, where we boost performance by 11%. This helps further highlight the benefits of our approach over prior general PR approaches.

4. Experiments on Large Language Models

Following standard protocols, we initialize LLaMA models with our CRISP reparameterization via neural mimicry and then fine-tune only the mixer matrices while keeping basis matrices frozen. We evaluate at two parameter budgets: an ultra-low regime with approximately 0.004% trainable parameters and a moderate regime at 0.01%, both substantially lower than conventional PEFT methods which typically use around 0.7-0.8% of base model parameters.

Tab. 5 demonstrates that CRISP is on par or better than its competitors with an order of magnitude fewer parameters. We also show that we can reduce the number of trainable parameters by another order of magnitude with only a minimal impact to performance. Note that prior work has shown PR methods like CRISP can also be composed with methods like HiRA and DoRA for further gains [13]. Further, most of these methods we compare to can only be applied to PEFT, whereas our approach can be used for compression as well, easing the implementation costs.

Algorithm 5 describes the LLaMA compression pipeline, which reduces basis rank via importance-weighted clustering and a short calibration stage using language modeling loss. Tab. 6 evaluates this against compression+PEFT combinations on commonsense reasoning at 30% parameter reduction. We follow the same procedure outlined in the main paper for adapter-finetuning and evaluation.

In the compression-only setting (main paper Tab. 3), CRISP outperforms all baselines by around 5 points - a gap consistent with the ViT results in Tab. 2 of the main paper. In the Tab. 6, we evaluate compressed models as initializations for downstream PEFT. CRISP with coefficient tuning achieves surpasses Basis-Sharing [15] and DFJR [10] combinations by a significant margin. PruneNet [12] combined with LoRA [4] or DoRA [7] achieves higher accuracy, though these combinations use substantially more trainable parameters during adaptation than CRISP's lightweight coefficient tuning. This highlights a parameter-accuracy tradeoff: CRISP provides a unified compress-once-adapt-freely framework at a lower adaptation cost, whereas pruning-based methods require a separate PEFT pipeline with a larger parameter budget to reach their best performance.

Table 3. Compression results on ViT-B/16 [2] with 50% weight compression across six diverse tasks (See Table 2). We find that using the distillation loss with SVD initialization described in Sec. 3.2 of our paper provides best performance.

Compression	Params(%)	Flowers	Aircraft	Scene	CFR100	CFR10	Birds	Avg
Neural Mimicry (Eq. 5 of main paper)	44M	83.9	56.3	51.2	49.8	79.4	26.3	57.8
Distillation Loss	44M	99.0	89.5	81.8	86.2	97.4	79.1	**88.8**
w/o SVD Init	44M	98.8	85.9	77.4	82.0	95.3	71.6	85.1

Table 4. Compression results on ViT-B/16 [2] at 75% parameter reduction evaluated on six fine-grained classification benchmarks (see Tab. 2 of the main paper for 50% reduction). **Upper section:** post-compression accuracy with only classifier adaptation. **Lower section:** MC+PEFT combinations demonstrate compressed models as initialization for downstream tasks. We find that CRISP outperforms prior work by up to 11%.

Compression	PEFT	Params(%)	Flowers	Aircraft	Scene	CFR100	CFR10	Birds	Avg
ViT-B/16 [2]	–	86M	96.7	70.9	84.5	76.3	97.0	84.6	85.0
SVD	–	21M	70.7	25.7	33.2	39.6	62.5	8.6	40.0
RECAST [13]	–	21M	90.0	59.1	67.2	67.1	85.3	52.9	70.2
CRISP (ours)	–	21M	95.1	73.0	77.7	74.1	89.7	73.6	**80.5**
SVD	Eigenvalues	21M	87.8	55.9	62.1	65.0	85.3	40.1	66.0
RECAST [13]	RECAST	21M	94.7	67.4	71.0	71.4	89.5	58.9	75.5
CRISP (ours)	CRISP (ours)	21M	98.8	85.2	79.6	82.6	95.9	75.1	**86.2**

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 3, 5

[3] Aaron Grattafiori and et al. The llama 3 herd of models, 2024. 6

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4, 6

[5] Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. HiRA: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 6

[6] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 3

[7] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024. 4, 6

[8] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. -, 2013. 3

[9] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 3

[10] Runyu Peng, Yunhua Zhou, Qipeng Guo, Yang Gao, Hang Yan, Xipeng Qiu, and Dahua Lin. Data-freeweight compress and denoise for large language models. *CoRR*, abs/2402.16319, 2024. 4, 6

[11] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. 3

[12] Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. You only prune once: Designing calibration-free model compression with policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 4, 6

[13] Nazia Tasnim and Bryan A. Plummer. Recast: Reparameterized, compact weight adaptation for sequential tasks. In *International Conference on Learning Representations (ICLR)*, 2025. 4, 5

[14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[15] Jingcun Wang, Yu-Guang Chen, Ing-Chao Lin, Bing Li, and Grace Li Zhang. Basis sharing: Cross-layer parameter sharing for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025. 4, 6

[16] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen,

Table 5. Comparison of PEFT methods on commonsense reasoning benchmarks. Results from LoRA and DoRA are taken from Liu et al. [7], HiRA results are from Huang et al. [5]. We find that CRISP is on par or better than custom PEFT methods while using an order of magnitude fewer parameters. Further, CRISP can also support MC, as we show in Tab. 6, demonstrating its ability to generalize to more PR tasks than prior PEFT methods.

Model	PEFT	Params(%)	Accuracy (↑)								
			BQ	PIQ	SIQ	Hell.	Win.	ARC-e	ARC-c	OBQ	Avg.
ChatGPT	—	—	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
Llama2-7B	LoRA [4]	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	$\text{DoRA}_{\text{half}}$ [7]	0.42	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	80.5
	DoRA [7]	0.84	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
	HiRA [5]	0.83	71.2	83.4	79.5	88.1	84.0	86.7	73.8	84.6	**81.4**
	CRISP	0.004	68.9	81.4	80.4	91.0	80.1	84.1	69.1	73.6	78.6
	CRISP	0.01	69.5	81.8	80.4	91.7	83.6	84.6	69.2	78.2	80.0
Llama3-8B	LoRA [4]	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	$\text{DoRA}_{\text{half}}$ [7]	0.36	74.5	88.8	80.3	95.5	84.7	90.1	79.1	87.2	85.0
	DoRA [7]	0.71	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	HiRA [5]	0.70	75.4	89.7	81.2	95.4	87.7	93.3	82.9	88.3	**86.7**
	CRISP	0.004	72.3	87.6	81.5	94.3	87.0	91.5	79.1	83.8	84.7
	CRISP	0.01	73.6	89.1	80.8	94.8	85.7	93.1	83.0	87.6	86.0

Table 6. LLaMA3.2-1B [3] MC+PEFT results at 30% parameter reduction across seven commonsense reasoning benchmarks. All baselines pair a compression method with a separate PEFT method, whereas CRISP compresses and adapts within the same factorized framework using lightweight coefficient tuning.

Compression	**PEFT**	Accuracy (↑)							
		BQ	PIQ	Hell.	Wino	ARC-e	ARC-c	OBQ	Avg.
LLaMA3.2-1B [3]	–	60.0	90.0	30.0	70.0	70.0	30.0	10.0	51.4
Basis-Sharing [15]	LoRA [4]	37.8	52.8	26.8	48.6	28.2	19.3	16.2	32.8
Basis-Sharing [15]	DoRA [7]	37.8	53.8	26.8	51.0	28.9	19.2	13.2	33.0
DFJR [10]	LoRA [4]	48.9	53.5	26.4	50.3	28.6	20.6	14.4	34.7
DFJR [10]	DoRA [7]	57.2	54.7	27.3	50.9	29.2	19.8	13.6	36.1
PruneNet [12]	LoRA [4]	62.4	57.8	32.3	54.6	39.4	22.1	16.8	40.8
PruneNet [12]	DoRA [7]	62.3	60.3	32.6	55.4	40.4	21.5	17.6	41.4
CRISP (ours)	CRISP (ours)	57.4	53.4	27.9	51.4	30.8	23.4	26.6	38.7

Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020. 2, 3