FourierMoE: Fourier Mixture-of-Experts Adaptation of Large Language Models

Juyong Jiang[1,2]* **Fan Wang**[1]* **Hong Qi**[1] **Sunghun Kim**[1] **Jing Tang**[1,2]†
[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] The Hong Kong University of Science and Technology
{csjuyongjiang,csfanwang,qinghecs}@gmail.com
{hunkim,jingtang}@ust.hk

Abstract

Parameter-efficient fine-tuning (PEFT) has emerged as a crucial paradigm for adapting large language models (LLMs). However, standard PEFT methods often struggle in multi-task fine-tuning settings due to task interference and a limited parameter budget. Recent approaches incorporate the mixture-of-experts (MoE) architecture, referred to as mixture-of-parameter-efficient-experts (MoPE), to alleviate this issue by dynamically routing inputs to specialized experts. However, these methods remain based on *spatial* parameterization, which may introduce structural redundancy and additional parameter overhead. To address these limitations, we revisit model adaptation from a *spectral* perspective. Our analysis uncovers heterogeneity in frequency sensitivity across model layers and downstream tasks, indicating that adaptation should be frequency-aware rather than uniformly parameterized in the spatial domain. Motivated by this insight, we propose **FourierMoE**, a novel framework that unifies spectral parameterization with MoE via frequency-specialized experts and the learning of conjugate-symmetric complex coefficients. We conduct extensive experiments across multiple model families on a diverse range of tasks, including commonsense reasoning, math reasoning, image classification, and natural language understanding. Experimental results on 28 benchmarks show that FourierMoE consistently outperforms full fine-tuning (FFT) and 16 competitive baselines in both single-task and multi-task settings, while requiring significantly fewer trainable parameters, demonstrating superior effectiveness, efficiency, and adaptability.

1 Introduction

The advent of large language models (LLMs) [19, 17, 68] has reshaped the landscape of natural language processing (NLP), yet their escalating scale renders the costs of full fine-tuning (FFT) prohibitive. To address this challenge, parameter-efficient fine-tuning (PEFT) [25, 38, 74] has emerged as a compelling paradigm. By updating a small set of parameters while keeping the backbone model frozen, PEFT substantially reduces the computational and memory overhead [35]. Despite their effectiveness in single-task adaptation, methods such as LoRA [25] often exhibit degraded performance in multi-task fine-tuning [33, 59]. This degradation arises from task interference and a limited parameter budget. In multi-task learning, distinct tasks may induce conflicting optimization objectives [69]. When diverse tasks are forced to share a single parameter space, their gradients can become orthogonal or even oppositional, leading to negative transfer [72, 37]. Furthermore, the restricted parameter budget of PEFT methods limits their capacity to model fine-grained, task-specific structures necessary for joint generalization across heterogeneous tasks [61, 38].

*Equal contributors. Listing order is random.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

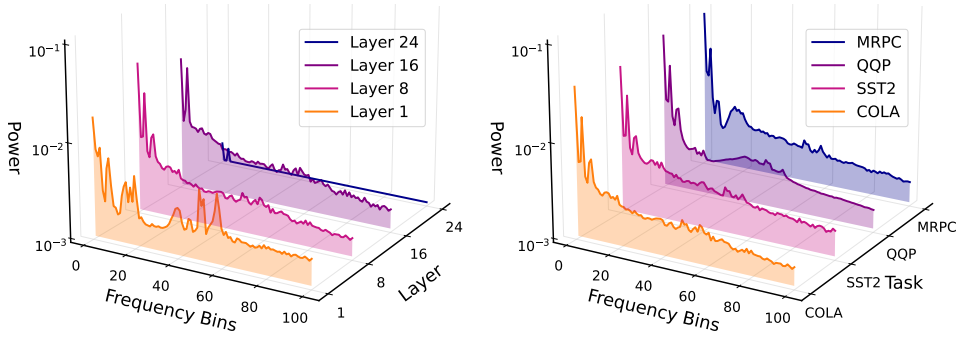

Figure 1: Spectral analysis on RoBERTa-large across layers and tasks. **(Left)** The power spectral density of the model weights shows layer-wise differences, with early layers exhibiting high-frequency spikes and deeper layers showing a smoother spectrum. **(Right)** For different GLUE tasks (CoLA, SST-2, QQP, and MRPC), the spectra of hidden representations from the model's eighth layer display distinct frequency energy distributions, revealing task-specific preferences.

The need to improve multi-task adaptability has motivated recent advances in mixture-of-parameter-efficient-experts (MoPE) approaches, which integrate the mixture-of-experts (MoE) architecture with PEFT [70, 10]. These approaches instantiate multiple lightweight PEFT modules as experts and employ a routing mechanism to conditionally activate task-relevant experts based on inputs [3, 4]. However, many existing MoPE methods [70, 33, 10] are still grounded in spatial parameterization, where adaptation capacity is scaled by adding multiple independently parameterized PEFT modules and relying on input-dependent routing to select among them. This design does not explicitly provide a specialization mechanism for encouraging functional diversity. Consequently, different experts may learn overlapping transformations, leading to structural redundancy [59, 15, 45]. Moreover, maintaining multiple experts introduces additional parameter overhead, weakening the efficiency that PEFT aims to achieve. These limitations suggest that simply adopting the MoE architecture is not fundamentally effective, highlighting the need for a more versatile and efficient framework for multi-task fine-tuning.

To address these limitations, we revisit model adaptation from a spectral perspective, where orthogonal Fourier bases provide a compact parameterization of weight updates [16, 47] and frequency decomposition offers a fine-grained lens [2, 12] for analyzing the adaptation requirements. We conduct a spectral analysis (see Figure 1), which reveals frequency sensitivity heterogeneity across model layers and downstream tasks. Specifically, early layers exhibit high-frequency spikes while deeper layers manifest a progressively smoother spectrum with attenuated high-frequency components, suggesting a transition from local feature extraction to global semantic integration. Task-specific energy distributions also emerge: MRPC exhibits an energy plateau across frequency bins 15–40, whereas SST-2 follows a smooth spectral decay. These observations imply that adaptation should be frequency-aware rather than uniformly parameterized in the spatial domain.

Motivated by these insights, we propose **FourierMoE** (**Fourier** **M**ixture-**o**f-**E**xperts), a novel MoPE method that unifies spectral parameterization with MoE. Our approach employs a frequency-adaptive router to dispatch inputs to the most relevant spectral experts, where each expert is assigned to a distinct frequency band and learns a small set of conjugate-symmetric complex coefficients. The selected spectral updates are transformed by inverse discrete Fourier transform (IDFT) into additive spatial-domain increments over the frozen backbone weights. This design addresses the limitations of prior methods through three dimensions: (1) FourierMoE facilitates specialization by assigning experts to distinct frequency bands, reducing task interference and mitigating structural redundancy. (2) Leveraging the compact expressivity of Fourier bases, our method learns a small number of spectral coefficients, substantially reducing parameter overhead compared with spatial MoPE methods. (3) We theoretically show that the asymmetric real-valued coefficients used in the prior spectral method [16] can induce representation error and information loss. FourierMoE, instead, learns both real and imaginary components under conjugate-symmetry constraints, preserving a more complete spectral representation while ensuring valid real-valued spatial reconstruction.

Extensive experimental results across NLP and computer vision (CV) domains show that FourierMoE achieves state-of-the-art (SOTA) results compared with competitive baselines in both single- and

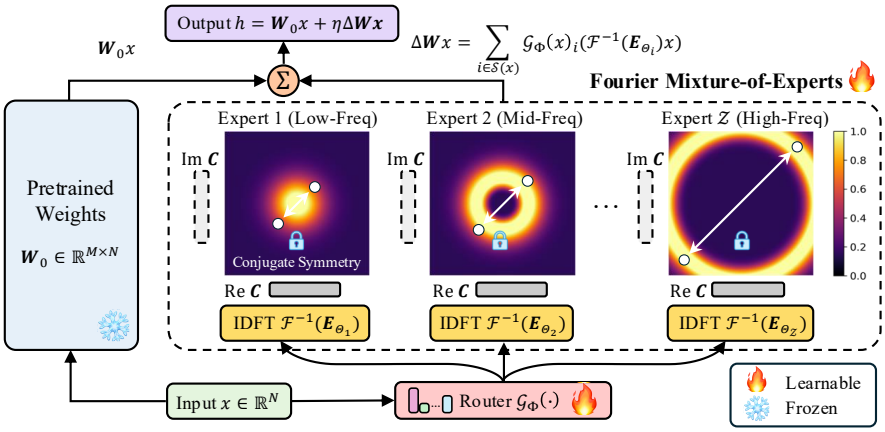

Figure 2: Overall framework of FourierMoE. Given an input x, the frequency-adaptive router $\mathcal{G}_\Phi(\cdot)$ dynamically selects the relevant experts from different frequency bands. The experts parameterize model weight updates $\Delta\mathbf{W}$ in the spectral domain by learning a small set of conjugate-symmetric complex coefficients. The selected spectral updates are transformed by IDFT into additive spatial increments over the frozen backbone weights.

multi-task setups [33], while using significantly fewer trainable parameters. Our main contributions are summarized as follows:

- We propose FourierMoE, a novel framework that unifies spectral parameterization with MoE. By assigning experts to distinct frequency bands and dynamically composing specialized updates, FourierMoE captures heterogeneous information to enhance adaptation performance, while mitigating structural redundancy and parameter overhead.
- Our theoretical analysis shows that existing spectral PEFT methods [16, 47] can suffer from information loss and representation error. To overcome these limitations, FourierMoE learns conjugate-symmetric complex coefficients, preserving spectral information while ensuring real-valued spatial updates. Combined with MoE, this design offers a new design direction for future spectral PEFT research.
- We conduct experiments across multiple model families and 28 benchmarks spanning commonsense reasoning, math reasoning, image classification, and natural language understanding (NLU). Results show that FourierMoE consistently outperforms FFT and 16 competitive baselines while requiring substantially fewer trainable parameters, demonstrating superior effectiveness and adaptability.

2 Methodology

In this section, we formally introduce FourierMoE, illustrated in Figure 2, which formulates model weight updates in the spectral domain. Motivated by the frequency heterogeneity observed in Figure 1, FourierMoE assigns experts to distinct frequency subspaces, where each expert learns conjugate-symmetric complex coefficients. The frequency-adaptive router dynamically composes input-relevant spectral updates. These selected spectral updates are mapped back to the spatial domain via IDFT and applied as additive residual updates to the frozen backbone weights. Our design provides structured expert specialization and enables efficient, frequency-aware adaptation.

2.1 Spectral Reparameterization

Conventional PEFT [25, 42, 38] and MoPE methods [10, 11, 70] typically learn weight updates $\Delta\mathbf{W}$ in the spatial domain. Given a frozen pretrained model $\mathbf{W}_0 \in \mathbb{R}^{M\times N}$, the adapted forward pass is $h = (\mathbf{W}_0 + \Delta\mathbf{W})x$. In contrast, FourierMoE parameterizes $\Delta\mathbf{W}$ in the spectral domain, where orthogonal Fourier bases provide a compact parameterization of weight updates [16, 47]. Specifically,

the spatial-domain update is obtained by applying IDFT to a learnable spectral matrix $\mathbf{F} \in \mathbb{C}^{M\times N}$:

$$\Delta\mathbf{W} = \mathcal{F}^{-1}(\mathbf{F}) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\mathbf{F}(u,v)\cdot\mathbf{B}_{u,v} \tag{1}$$

where $\mathbf{B}_{u,v} = \exp\big(j2\pi\left(\frac{uq}{M}+\frac{vy}{N}\right)\big)$ is the 2D Fourier basis kernel. (u,v) indexes a spectral coordinate, (q,y) indexes the reconstructed spatial matrix entry, with $u,q \in \{0,\dots,M-1\}$ and $v,y \in \{0,\dots,N-1\}$. $\mathbf{F}(u,v)$ is the spectral coefficient. This spectral parameterization offers a compact yet expressive representation of weight updates. Beyond compact parameterization, the Fourier representation provides an explicit mechanism for decomposing weights into frequency components. This decomposition enables FourierMoE to assign experts to distinct frequency subspaces, providing a structured basis for expert specialization.

2.2 Frequency-Specialized Experts

In the spectral domain, signals can be decomposed into various frequency components [12, 67], providing a natural basis for expert specialization. FourierMoE implements frequency-wise specialization by assigning each expert a distinct set of spectral coordinates. Specifically, the distance $\mathcal{D}(u,v)$ of a coordinate (u,v) to the DC component (origin) determines the corresponding frequency scale [46]. This coordinate-based assignment can constrain experts to operate within distinct frequency subspaces, such as low-frequency regions near the origin or higher-frequency regions farther away.

Formally, the specialization of an expert $\mathbf{E}_i \in \mathbb{C}^{M\times N}$ is determined by the frequency band covered by its assigned coordinate set Ω_i. We construct Ω_i using a Gaussian bandpass filter [18]. For each spectral coordinate (u,v), its assignment probability to expert $\mathbf{E}_i$ is defined as:

$$P_i(u,v) = \exp\left(-\left(\frac{\mathcal{D}(u,v)^2 - \mathcal{D}_{c,i}^2}{\mathcal{D}(u,v)\cdot\mathcal{W}_i}\right)^2\right) \tag{2}$$

where $\mathcal{D}_{c,i}$ is the center frequency of the i-th band, and $\mathcal{W}_i$ controls its bandwidth. Based on $P_i(u,v)$, each expert is assigned a subset of frequency coordinates concentrated around its designated center frequency $\mathcal{D}_{c,i}$. This band-limited construction encourages FourierMoE to capture information ranging from global semantics (low-frequency) to local syntactic variations (high-frequency), enhancing generalization capability across multiple tasks.

2.3 Conjugate-Symmetric Complex Coefficients

The informational signals within neural network weights exhibit spectral sparsity [47], with important information concentrated in a subset of dominant frequency components [58, 12]. In addition, FourierFT empirically shows that sparse spectral coefficients can recover high-quality weight updates [16]. These findings support utilizing a sparse spectral subspace for parameterizing $\Delta\mathbf{W}$, where task-specific weight updates are learned by optimizing a limited number of spectral coefficients. Accordingly, each Fourier expert $\mathbf{E}_i \in \mathbb{C}^{M\times N}$ learns complex coefficients $\mathbf{C}_i$ at n frequency coordinates from Ω_i, where $n \ll MN$. FourierMoE differs from existing spectral PEFT methods [16, 47] by enforcing conjugate-symmetry constraints and adopting a complex-valued representation.

2.3.1 Conjugate-Symmetric Constraint

Prior spectral PEFT methods [16, 47] learn spectral weight updates $\mathbf{F}_{\text{unsym}}$ from unconstrained, asymmetric spectral coefficients. We next show that such asymmetry can introduce information loss.

Theorem 2.1 (Conjugate Symmetry Condition). *In FourierMoE, given that the backbone model $\mathbf{W}_0$ and the input x are real-valued, the resulting update $\Delta\mathbf{W}$ must lie in $\mathbb{R}^{M\times N}$. To guarantee real-valued updates independently of the routing weights, each expert update should be real-valued. Let $\mathbf{F} \in \mathbb{C}^{M\times N}$ be a spectral matrix and let $\mathbf{S} = \mathcal{F}^{-1}(\mathbf{F})$. Then $\mathbf{S} \in \mathbb{R}^{M\times N}$ if and only if $\mathbf{F}$ satisfies Hermitian symmetry:*

$$\mathbf{F}(u,v) = \mathbf{F}^*(\langle -u\rangle_M, \langle -v\rangle_N), \quad \forall(u,v), \tag{3}$$

where $\langle\cdot\rangle_M$ denotes the modulo M operation and $(\cdot)^*$ denotes the complex conjugate.

Proof. For each spatial coordinate (q, y), the IDFT is:

$$\mathbf{S}(q,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathbf{F}(u,v) e^{j\frac{2\pi uq}{M}} e^{j\frac{2\pi vy}{N}} \tag{4}$$

Let $\theta_{u,v} = 2\pi(\frac{uq}{M} + \frac{vy}{N})$. We split the summation into the DC component, the Nyquist components (if any), and pairs of indices (u, v) and their reflection $(u', v') = (\langle -u \rangle_M, \langle -v \rangle_N)$. Assume the symmetry condition holds: $\mathbf{F}(u', v') = \mathbf{F}^*(u, v)$. The contribution of this pair to the sum is:

$$T = \mathbf{F}(u,v)e^{j\theta_{u,v}} + \mathbf{F}(u',v')e^{j\theta_{u',v'}} \tag{5}$$

$$= \mathbf{F}(u,v)e^{j\theta_{u,v}} + \mathbf{F}^*(u,v)e^{j(2\pi k - \theta_{u,v})} \, (k \in \mathbb{Z}) \tag{6}$$

$$= \mathbf{F}(u,v)e^{j\theta_{u,v}} + \left(\mathbf{F}(u,v)e^{j\theta_{u,v}}\right)^* \tag{7}$$

$$= 2\,\mathrm{Re}\left\{\mathbf{F}(u,v)e^{j\theta_{u,v}}\right\}. \tag{8}$$

Since $2\,\mathrm{Re}\{\cdot\}$ is strictly real, the total sum $\mathbf{S}(q, y)$ comprises only the real terms. Conversely, if symmetry is violated, $\mathbf{S}$ cannot be real-valued, resulting in at least one $\mathbf{S}(q, y) \in \mathbb{C}$. □

Therefore, the learning of asymmetric spectral coefficients, as done in prior works, cannot guarantee real-valued spatial updates. These methods obtain spatial-domain updates by retaining only the real part of the inverse transform, i.e., $\Delta\mathbf{W}_{\text{trunc}} = \mathrm{Re}\left(\mathcal{F}^{-1}(\mathbf{F}_{\text{unsym}})\right)$. Such an operation can result in truncation errors and information loss.

Corollary 2.2 (**Truncation Error Bound**). *Let $\mathbf{F}_{unsym}$ be a spectral matrix violating Theorem 2.1. The effective weight update for the model is $\mathbf{W}_{eff} = \mathrm{Re}(\mathcal{F}^{-1}(\mathbf{F}_{unsym}))$. The information loss due to the truncation of imaginary parts, defined as the energy of the discarded signal, is given by:*

$$\mathcal{L}_{error} = \|\mathcal{F}^{-1}(\mathbf{F}_{unsym}) - \mathbf{W}_{eff}\|_F^2 = \sum_{q,y} \left(\mathrm{Im}(\mathcal{F}^{-1}(\mathbf{F}_{unsym}))_{q,y}\right)^2.$$

By Parseval's theorem [46], the truncation error can be equivalently characterized in the spectral domain, up to a DFT-dependent normalization constant:

$$\mathcal{L}_{error} \propto \sum_{u,v} \|\mathbf{F}_{unsym}(u,v) - \mathbf{F}^*_{unsym}(\langle -u \rangle_M, \langle -v \rangle_N)\|^2. \tag{9}$$

Without enforcing conjugate symmetry, reconstructed updates that fall into the imaginary space are discarded, causing information loss. FourierMoE enforces the constraint in Theorem 2.1 for each expert $\mathbf{E}_i$ during the learning of coefficients, ensuring the update contributed by each expert is strictly real-valued, reducing the truncation error.

2.3.2 Complex-Valued Representation

In the spectral domain, a coefficient located at a specific coordinate is represented as $\mathbf{C}(u, v) = a_{u,v} + jb_{u,v}$, where $a_{u,v}$ and $b_{u,v}$ denote its real and imaginary components, and j is the imaginary unit satisfying $j^2 = -1$. However, the prior method [16] instantiates its learnable coefficients as real-valued, which may introduce representation errors.

Proposition 2.3 (**Phase-Amplitude Completeness**). *Restricting spectral coefficients to $\mathbb{R}$ (i.e., $b_{u,v} = 0$) forces the phase $\Phi_{u,v} = \mathrm{atan2}(b, a)$ to be either 0 or π. This constrains the resulting spatial signal to be an even function (symmetric around the origin), rendering the model incapable of representing spatial shifts or asymmetric features in $\Delta\mathbf{W}$. By learning $\mathbf{C} \in \mathbb{C}$, FourierMoE maintains full expressivity over amplitude $A = \sqrt{a^2 + b^2}$ and phase Φ.*

2.4 Frequency-Adaptive Router

Given a token representation $x \in \mathbb{R}^N$, the router computes a distribution over Z experts:

$$g(x) = \mathrm{softmax}(\mathbf{W}_g x), \tag{10}$$

$$\mathcal{S}(x) = \{i \mid g(x)_i \text{ is among the top-}K \text{ entries of } g(x)\}, \tag{11}$$

$$\mathcal{G}_\Phi(x)_i = \begin{cases} g(x)_i, & i \in \mathcal{S}(x), \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathbf{W}_g \in \mathbb{R}^{Z\times N}$ is the router weight matrix and K is the number of selected experts per token. The final composite update $\Delta\mathbf{W}$ is:

$$\Delta\mathbf{W} = \sum_{i\in\mathcal{S}(x)} \mathcal{G}_\Phi(x)_i \cdot \Delta\mathbf{W}_i, \quad \Delta\mathbf{W}_i = \mathcal{F}^{-1}(\mathbf{E}_i). \tag{13}$$

We emphasize that, as in standard sparse MoE architectures such as Switch Transformer [13], GShard [32], and Mixtral 8×7B [27], the routing is token-wise in definition but batch-parallel in execution. For a batch of tokens, router scores are computed jointly through batched matrix multiplications, and routed tokens are then grouped by expert for parallel expert computation. FourierMoE follows the same execution paradigm. In particular, for each forward pass, the spatial update $\Delta\mathbf{W}_i = \mathcal{F}^{-1}(\mathbf{E}_i)$ of an activated expert is reconstructed at most once per layer and reused for all tokens assigned to that expert. Therefore, FourierMoE preserves the efficient batched execution paradigm of standard MoE models, with the additional reconstruction overhead incurred only once per activated expert rather than once per token. Empirical results in Table 7 confirm that this design remains efficient in practice.

2.5 Optimization Objective

The forward pass for the l-th layer is formulated as:

$$h = \mathbf{W}_0 x + \eta \sum_{i\in\mathcal{S}(x)} \mathcal{G}_\Phi(x)_i \left(\mathcal{F}^{-1}(\mathbf{E}_{\Theta_i})x\right), \tag{14}$$

where η denotes a predefined scaling factor, $\Theta = \{\Theta_i\}_{i=1}^{Z}$ represents the ensemble of learnable spectral coefficients for all experts, and Φ denotes the trainable parameters of the frequency-adaptive router $\mathcal{G}$. We optimize the joint parameter set $\{\Theta, \Phi\}$ by minimizing a composite objective function that balances task performance with architectural stability. Formally, the optimization problem is defined as the minimization of the expected loss over the training distribution $\mathcal{D}_{train}$:

$$\min_{\Theta,\Phi} \mathbb{E}_{x\sim\mathcal{D}_{train}} \left[\mathcal{L}_{task}(x;\Theta,\Phi) + \lambda\mathcal{L}_{aux}(x;\Phi)\right], \tag{15}$$

where $\mathcal{L}_{task}$ is the standard autoregressive cross-entropy loss. To mitigate the risk of expert collapse, a common failure mode in sparse MoE where the router over-selects a small subset of experts, we define the auxiliary load-balancing loss $\mathcal{L}_{aux}$ following prior works [32, 13]:

$$\mathcal{L}_{aux} = Z\sum_{i=1}^{Z} f_i \cdot P_i, \quad f_i = \frac{1}{B}\sum_{x\in\mathcal{B}} \mathbf{1}\{i \in \mathcal{S}(x)\} \tag{16}$$

where f_i is the fraction of tokens in a batch $\mathcal{B}$ assigned to expert i (i.e., expert i is in the top-K set for x), and P_i is the mean routing probability for expert i across the batch. This formulation balances expert optimization for broad spectral coverage and diverse representations. For clarity, we present the detailed algorithm in Appendix A.

3 Experiments

3.1 Experiment Setting

Baselines. We compare FourierMoE against FFT, PEFT, and MoPE baselines. For a fair and comprehensive evaluation, we follow the configurations used in prior works and reuse their reported results. **PEFT** baselines include LoRA [25], rsLoRA [28], DoRA [38], PiSSA [42], MiLoRA [64], KaSA [63], LoRA-Dash [55], NEAT [75], TopLoRA [34], MELoRA [51], LFMA[47], and FourierFT [16]. **MoPE** baselines include MoLoRA [70], AdaMoLE [40], HydraLoRA [59], and GOAT [11]. The detailed introduction of baselines is presented in Appendix C.

Models and Datasets. We evaluate the efficiency of FourierMoE across multi-task and single-task scenarios spanning NLP and computer vision (CV) domains. For the **multi-task** setup [33], we train models on mixed tasks and evaluate them on the individual test set, including: (1) Commonsense reasoning: we fine-tune LLaMA-2 7B [60] and Gemma 7B [17] on the Commonsense170K dataset [26], and evaluate on the test sets of its constituent subsets. (2) Math reasoning: we fine-tune LLaMA-3 8B [19] and Qwen2.5-14B [49] on Math10K [26], and evaluate them on GSM8K [9], SVAMP [48],

Table 1: Performance comparison of PEFT and MoPE methods on eight commonsense reasoning benchmarks using LLaMA-2 7B and Gemma 7B in a multi-task setting. Accuracy is reported for all benchmarks. **Bold** denotes the best result.

Model	Method	# Params(%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
ChatGPT	/	/	73.10	85.40	68.50	78.50	66.10	89.80	79.90	74.80	77.01
LLaMA-2 7B	LoRA	0.84	69.80	79.90	79.50	83.60	82.60	79.80	64.70	81.00	77.61
	DoRA	0.84	71.80	83.10	79.90	89.10	83.00	84.50	71.00	81.20	80.45
	PiSSA	0.84	67.60	78.10	78.40	76.60	78.00	75.80	60.20	75.60	73.79
	MiLoRA	0.84	67.60	83.80	80.10	88.20	82.00	82.80	68.80	80.60	79.24
	LoRA-Dash	0.84	71.00	75.70	79.30	91.10	78.60	84.20	69.80	78.80	78.56
	NEAT	0.84	71.70	83.90	80.20	88.90	84.30	86.30	71.40	83.00	81.21
	KaSA	0.84	73.60	84.40	80.20	91.50	84.50	84.70	72.10	81.20	81.53
	HydraLoRA	0.84	72.78	84.06	79.68	80.34	86.66	87.12	72.35	86.00	81.12
	MoLoRA	0.96	73.15	83.68	80.09	74.57	85.95	87.33	72.53	86.20	80.43
	GOAT	0.96	73.60	83.95	80.50	87.12	85.00	87.79	76.88	87.00	82.73
	FourierMoE	0.06	**73.73**	**84.60**	**81.27**	**92.33**	**86.82**	**88.24**	**77.21**	**87.40**	**83.95**
Gemma 7B	LoRA	0.14	75.17	88.74	77.58	95.21	89.11	92.93	84.73	88.00	86.43
	DoRA	0.14	73.49	90.44	79.19	95.03	90.00	93.79	84.73	88.67	86.92
	MELoRA	0.14	73.49	89.50	79.99	94.60	89.90	93.18	84.30	89.40	86.79
	HydraLoRA	0.16	72.14	89.21	81.30	95.11	89.45	94.56	85.32	89.00	87.01
	FourierMoE	0.03	**75.26**	**91.05**	**82.55**	**95.81**	**90.21**	**95.44**	**85.62**	**89.60**	**88.19**

Table 2: Performance comparison of PEFT and MoPE methods on six math reasoning benchmarks using LLaMA-3 8B and Qwen2.5-14B in a multi-task setting. Accuracy is reported for all benchmarks. **Bold** denotes the best result.

Model	Method	# Params(%)	AddSub	AQuA	GSM8K	MultiArith	SingleEQ	SVAMP	Average
LLaMA-3 8B	LoRA	0.12	84.56	25.72	57.22	91.22	92.26	70.17	70.19
	DoRA	0.12	85.95	26.19	56.52	89.67	**92.62**	70.40	70.23
	MELoRA	0.12	85.82	24.41	55.34	87.83	91.54	71.20	69.36
	HydraLoRA	0.11	86.08	25.98	55.50	91.00	91.14	68.10	69.63
	FourierMoE	0.01	**87.85**	**33.07**	**60.80**	**93.16**	91.54	**73.00**	**73.24**
Qwen2.5-14B	LoRA	0.12	91.90	34.65	74.37	96.33	92.91	86.40	79.43
	DoRA	0.13	92.41	36.22	75.92	96.39	92.39	86.80	80.02
	TopLoRA	0.10	91.31	35.96	77.31	**97.67**	93.44	87.43	80.52
	MELoRA	0.12	92.91	33.86	75.89	97.33	92.13	85.60	79.62
	HydraLoRA	0.12	92.41	36.61	76.32	96.22	92.45	86.97	80.16
	FourierMoE	0.05	**94.68**	**41.34**	**78.09**	97.33	**95.87**	**91.80**	**83.19**

MultiArith [52], AddSub [24], AQuA [36], and SingleEQ [29]. For the **single-task** setup, we train and evaluate models on each task, including: (3) Image classification: we fine-tune and evaluate CLIP ViT-B/32 [50] on Cars [30], DTD [6], EuroSAT [23], GTSRB [56], RESISC45 [5], SUN397 [66], and SVHN [44]. (4) Natural language understanding: we fine-tune and evaluate RoBERTa-large [39] on the CoLA, SST-2, MRPC, QQP, MNLI, QNLI, and RTE subsets from GLUE [62]. We present detailed statistics and hyperparameter configurations for all datasets and benchmarks in Appendix D and Appendix E. All experiments are conducted on NVIDIA H100 (80GB) GPUs.

3.2 Main Results

Commonsense Reasoning. As shown in Table 1, FourierMoE consistently surpasses the baselines across all benchmarks and models, while using minimal trainable parameters. On LLaMA-2 7B, it outperforms the strongest MoPE baseline, GOAT, by 1.22 percentage points (pp), while using 16$\times$ fewer trainable parameters. On Gemma 7B, our method achieves the best performance of 88.19 with 0.03% of parameters, demonstrating its superior efficiency in the multi-task setup.

Math Reasoning. We evaluate FourierMoE on larger LLMs to validate its scalability. Results in Table 2 show that our method consistently attains the highest average accuracy across both backbone models. On LLaMA-3 8B, it improves accuracy by 3.01 pp over the strongest PEFT baseline (DoRA) and by 3.61 pp over the best MoPE method (HydraLoRA). On Qwen2.5-14B, FourierMoE achieves SOTA across 5/6 benchmarks, demonstrating robust scalability across models.

Image Classification. Table 3 demonstrates that FourierMoE surpasses all PEFT and MoPE baselines across CV benchmarks. It outperforms FFT by 2.85 pp and FFT MoE by 0.7 pp while using 0.42% parameters, significantly lowering adaptation costs. The consistent improvements underscore the versatility and cross-modal generalizability of FourierMoE beyond NLP tasks. Our method consistently outperforms the spatial MoPE baselines, including MoLoRA, AdaMoLE, HydraLoRA, and GOAT, indicating the effectiveness of the spectral reparameterization for MoE.

Table 3: Performance comparison of FFT, PEFT, and MoPE methods on seven image classification benchmarks using CLIP ViT-B/32 in a single-task setting. Accuracy is reported for all benchmarks. **Bold** denotes the best result.

Method	# Params (%)	Cars	DTD	EuroSAT	GTSRB	RESISC45	SUN397	SVHN	Average
FFT	100	60.33	73.88	98.96	98.30	93.65	53.84	96.78	82.25
FFT MoE	770	66.39	75.53	98.59	98.50	94.38	60.34	97.09	84.40
LoRA (rank8)	1.49	41.02	70.15	98.66	96.51	90.38	47.51	95.39	77.09
LoRA (rank16)	2.99	46.51	72.07	98.74	98.04	92.08	51.63	96.00	79.30
LoRA (rank32)	5.98	50.13	72.87	98.88	98.13	92.87	53.65	96.55	80.44
DoRA	1.49	40.75	71.91	98.89	97.71	90.19	47.54	95.46	77.49
PiSSA	1.49	40.41	69.62	98.48	95.84	90.58	47.21	95.84	76.85
MiLoRA	1.49	39.77	70.48	98.19	97.52	89.92	45.38	95.49	76.68
FourierFT	0.22	39.20	70.58	98.02	97.93	91.58	61.92	91.14	78.62
MoLoRA	2.24	50.83	73.51	98.63	97.72	92.58	52.55	96.00	80.26
AdaMoLE	2.33	49.47	71.65	98.52	97.73	91.95	52.29	95.82	79.63
HydraLoRA	1.58	48.42	72.18	98.40	97.28	92.93	51.80	96.06	79.58
GOAT	2.24	53.50	75.32	98.82	98.17	93.46	54.53	96.62	81.49
FourierMoE	0.42	**64.44**	**77.18**	**98.93**	**98.28**	**93.65**	**66.41**	**96.78**	**85.10**

Table 4: Performance comparison of FFT, PEFT, and MoPE methods on seven benchmarks from GLUE using RoBERTa-large in a single-task setting. Accuracy is reported for all benchmarks. **Bold** denotes the best result.

Method	# Params (%)	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	Average
FFT	100	84.27	95.98	85.29	91.58	89.83	94.49	84.84	89.47
FFT MoE	698	86.02	96.22	85.05	92.20	90.20	95.10	84.48	89.90
LoRA	4.00	83.41	95.64	83.33	90.06	89.00	93.28	84.47	88.46
DoRA	4.00	85.33	95.99	84.07	91.24	89.52	93.54	84.48	89.17
PiSSA	4.00	69.12	95.98	82.84	91.24	88.94	93.59	73.29	85.00
MiLoRA	4.00	84.65	96.10	86.02	91.33	89.51	94.12	84.83	89.51
rsLoRA	4.00	83.51	95.98	86.02	90.75	88.97	93.84	84.12	89.03
FourierFT	0.42	78.01	95.20	86.27	85.48	87.19	91.59	83.39	86.73
LFMA	0.42	84.28	**96.56**	91.18	91.29	89.66	94.23	85.20	90.34
MoLoRA	4.50	83.94	96.10	87.75	91.45	89.36	93.90	84.11	89.52
AdaMoLE	4.56	83.99	95.76	86.03	91.48	89.21	93.64	83.75	89.12
GOAT	4.50	**86.86**	96.21	84.55	91.40	89.55	94.19	85.56	89.76
HydraLoRA	2.75	83.89	95.52	85.04	91.02	89.34	93.87	81.22	88.56
FourierMoE	0.42	85.52	96.33	**91.91**	**91.52**	**90.33**	**94.66**	**89.53**	**91.40**

Natural Language Understanding. Table 4 shows that our method outstrips PEFT and MoPE baselines across 5/7 benchmarks using 0.42% of trainable parameters. It exceeds the best PEFT baseline, LFMA (90.34), by 1.06 pp, and the best MoPE method, GOAT (89.76), by 1.64 pp. FourierMoE also outperforms FourierFT (91.40 vs. 86.73), another spectral PEFT method, under the same parameter budget. These outcomes suggest the efficiency of our method in the single-task fine-tuning scenario.

3.3 In-depth Analysis and Insights

Ablation Study. We assess the contribution of each component of FourierMoE across the Cars, DTD, and SUN397 benchmarks using CLIP ViT-B/32. Specifically, the w/o imaginary part variant learns only the real component of the Fourier coefficients, while the w/o real part variant learns only the imaginary component. The w/o frequency bias variant randomly samples the frequency coordinates for each expert. The w/o conjugate symmetry variant removes the conjugate symmetry constraints. As shown in Figure 3, the full implementation of FourierMoE consistently achieves the best performance across the evaluated benchmarks, whereas ablating any component leads to performance degradation, validating the effectiveness of the core design choices.

Expert Scalability. We evaluate the scalability of FourierMoE in terms of number of coefficients, number of experts, and activation count on MRPC, QNLI, and RTE using RoBERTa-large. As illustrated in Figure 4 (left), the model performance initially scales with the coefficient count and peaks at $n = 1008$, beyond which further increases lead to performance degradation. A similar trend is observed for the total number of experts Z (center), where accuracy improves up to $Z = 8$ before declining. Regarding the expert activation count K (right), $K = 2$ provides a favorable balance between performance and sparsity across the evaluated tasks, while excessively sparse or dense

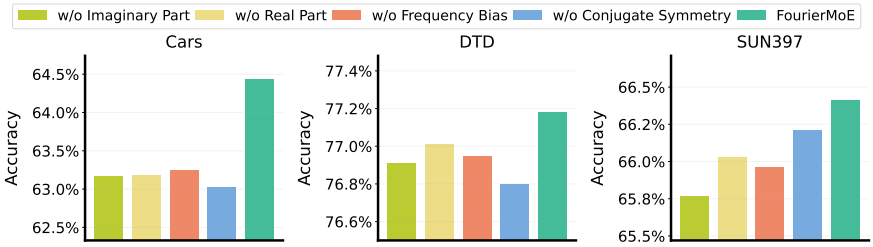

Figure 3: Component ablation study of FourierMoE on Cars, DTD, and SUN397. Removing any component results in a performance drop compared to the full FourierMoE implementation.

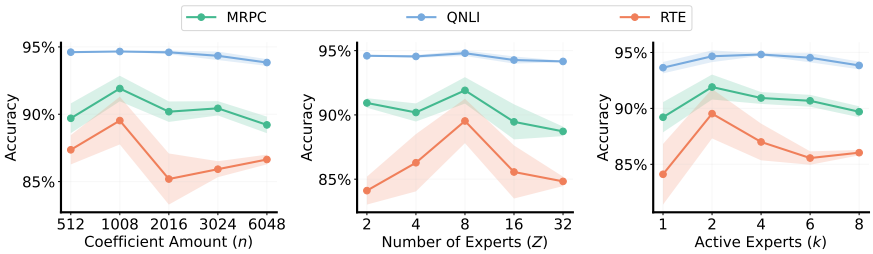

Figure 4: Expert scaling analysis on MRPC, QNLI, and RTE. **Left:** Impact of the trainable coefficient count n per expert. **Center:** Scalability with respect to the total number of experts Z (fixed activation $K = 2$). **Right:** Effect of the active expert count K given a fixed total pool size of $Z = 8$.

Table 5: Performance comparison of FourierMoE under different frequency bandwidths ($\mathcal{W}$) on CoLA, MRPC, and RTE.

$\mathcal{W}$	CoLA	MRPC	RTE
0.06	83.03	90.44	86.28
0.12	85.52	91.91	89.53
0.24	83.89	89.71	87.36
0.48	83.31	88.48	86.28
0.96	82.74	88.95	84.47

Table 6: Performance comparison of FourierMoE using various load balancing loss weight (λ) on Cars, DTD, and SUN397.

λ	Cars	DTD	SUN397
0.001	64.44	77.18	66.41
0.005	64.32	77.44	65.88
0.01	64.25	77.12	66.00
0.05	64.00	76.91	65.96
0.1	64.32	76.80	65.81

activation tends to be suboptimal. These findings indicate that FourierMoE balances performance with structural sparsity, achieving high parameter efficiency.

Hyperparameter Sensitivity. We evaluate the impact of the frequency bandwidth $\mathcal{W}$ and load balancing loss weight λ. As shown in Table 5, expanding $\mathcal{W}$ from 0.06 to 0.12 improves performance across benchmarks, while a further increase beyond this point causes accuracy degradation. We speculate that this is related to the role of $\mathcal{W}$ in determining the specialization of experts, consequently shaping model performance. We provide a detailed analysis of frequency bandwidth in Appendix F.1. Furthermore, Table 6 shows that variations in λ lead to marginal fluctuations in performance. This suggests that FourierMoE maintains a stable optimization trajectory, indicating that the auxiliary routing regularization does not significantly interfere with the task learning objective.

Computation Efficiency. Table 7 compares the efficiency of FourierMoE with PEFT baselines (LoRA and TopLoRA) and a MoPE baseline (GOAT). FourierMoE uses only 1.49M trainable parameters, which is 37% fewer than LoRA (2.38M), 37% fewer than TopLoRA (2.36M), and 11.4× fewer than GOAT (16.98M), while achieving the best performance (89.53), improving over LoRA, TopLoRA, and GOAT by +7.22, +4.33, and +3.97 pp, respectively. In terms of latency, FourierMoE is 15× faster than GOAT during training (58 s vs. 867 s) and 2.7× faster during inference (313 ms vs. 840 ms), although it remains slower than the standard PEFT baseline LoRA due to the routing and IDFT operations. These results show that FourierMoE achieves a better efficiency-performance trade-off than the three strong baselines.

Expert Frequency. Figure 5 visualizes the expert-wise spectral coordinates across different layers of CLIP ViT-B/32 fine-tuned on EuroSAT. Different experts progressively occupy frequency bands

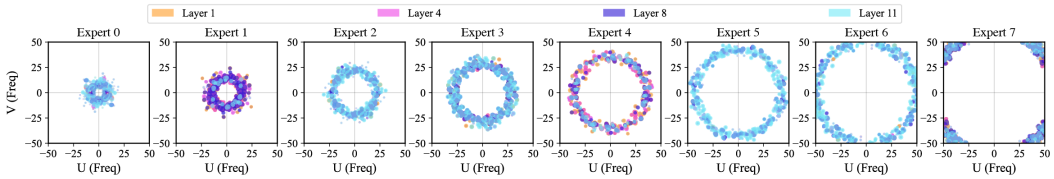

Figure 5: Expert-wise distributions of spectral coordinates in CLIP ViT-B/32 fine-tuned on EuroSAT.

Table 7: Efficiency comparison of methods fine-tuning RoBERTa-large on RTE using a single NVIDIA H100 (80GB) GPU. Note that training FLOPs ($\times 10^9$) are calculated per sample, training latency is reported per epoch, and inference latency is measured with a batch size of 32.

Metric	**LoRA**	**TopLoRA**	**GOAT**	**FourierMoE**
# Trainable Params	2.38 M	2.36 M	16.98 M	1.49 M
# GPU Memory	12.24 GB	16.90 GB	17.12 GB	11.61 GB
# Training FLOPs	340.02	258.84	163.46	235.23
Training Latency	13.13 s	145.69 s	867 s	58 s
Inference Latency	53.4 ms	397 ms	840 ms	313 ms
Performance	82.31	85.20	85.56	89.53

Table 8: Expert-wise token routing patterns in LLaMA-2 7B fine-tuned on BoolQ. For each expert, we report its center frequency (D_c) and the five most frequently routed tokens.

Expert ID	**Center Frequency** (D_c)	**Representative Tokens**
Expert 0	0.185	`can`, `the`, `in`, `are`, `did`
Expert 1	0.362	`a`, `an`, `is`, `that`, `and`
Expert 2	0.508	`the`, `Below`, `Write`, `response`, `following`
Expert 3	0.609	`from`, `for`, `end`, `dead`, `come`
Expert 4	0.698	`Response`, `at`, `going`, `after`, `girl`
Expert 5	0.789	`question`, `false`, `answer`, `there`, `same`
Expert 6	0.880	`format`, `season`, `cup`, `movie`, `sweet`
Expert 7	0.944	`of`, `you`, `do`, `be`, `true`

ranging from low-frequency regions to increasingly higher-frequency annular regions, while the distributions for the same expert cluster within a coherent spectral region across layers. This suggests that FourierMoE induces distinct frequency subspaces across experts, helping reduce structural overlap and enabling the router to compose spectrally diverse updates for heterogeneous inputs.

Expert Specialization. We conduct a specialization analysis on the functionality of experts by examining token preferences of experts in LLaMA-2 7B fine-tuned on BoolQ. Table 8 shows that tokens with different complexities are routed to experts with different frequency bands. Specifically, low-frequency experts are more frequently activated by common functional tokens (e.g., "the", "and"), whereas high-frequency experts preferentially process task-related content tokens (e.g., "question", "answer"), indicating functional specialization. In addition, our weight-level similarity analysis in Appendix F.2 reveals negligible overlap among expert weights. These results indicate that FourierMoE promotes meaningful expert specialization and structurally avoids redundancy among expert weights.

4 Conclusion

In this work, we first conduct a spectral analysis that reveals heterogeneity in frequency sensitivity across both model layers and downstream tasks, suggesting the necessity of fine-grained adaptation in the spectral domain. Based on these observations, we propose FourierMoE, a frequency-aware MoPE framework for adapting LLMs. Specifically, FourierMoE departs from spatial-domain adaptation by routing tokens to experts specialized in distinct frequency bands, with each expert learning conjugate-symmetric complex coefficients in the Fourier domain. This design helps reduce inter-expert redundancy and task interference under limited parameter budgets, while encouraging diverse frequency representations and theoretically ensuring real-valued weight updates. Extensive evaluations across 28 benchmarks show that FourierMoE consistently outperforms full fine-tuning and 16 competitive PEFT and MoPE baselines with superior parameter efficiency, highlighting the potential of spectral modulation for scalable LLM adaptation.

Acknowledgments

This work was partially supported by National Key R&D Program of China under Grant No. 2024YFA1012700, by the National Natural Science Foundation of China (NSFC) under Grant No. 62402410, by Guangdong Provincial Project (No. 2023QN10X025), by the Guangzhou Science and Technology Program (Grant No. 2026C99J0005), and by HKUST(GZ) Kunpeng&Ascend Center of Cultivation.

References

[1] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[2] Shubhankar Borse, Shreya Kadambi, Nilesh P Pandey, Kartikeya Bhardwaj, Viswanath Ganapathy, Sweta Priyadarshi, Risheek Garrepalli, Rafael Esteves, Munawar Hayat, and Fatih Porikli. Foura: Fourier low-rank adaptation. *Advances in Neural Information Processing Systems*, 37:71504–71539, 2024.

[3] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024.

[4] Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160*, 2024.

[5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979*, 4(7), 2023.

[11] Chenghao Fan, Zhenyi Lu, Sichen Liu, Chengfeng Gu, Xiaoye Qu, Wei Wei, and Yu Cheng. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment. *arXiv preprint arXiv:2502.16894*, 2025.

[12] Yijia Fan, Jusheng Zhang, and Keze Wang. Towards more efficient post-training via fourier domain adapter framework. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6175–6193, 2025.

[13] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[14] Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. Mixture-of-loras: An efficient multitask tuning for large language models. *arXiv preprint arXiv:2403.03432*, 2024.

[15] Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. Higher layers need more lora experts. *arXiv preprint arXiv:2402.08562*, 2024.

[16] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

[17] Gemma Team. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[18] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[20] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

[21] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[22] Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. Fourier transformer: Fast long range modeling by removing sequence redundancy with fft operator. *arXiv preprint arXiv:2305.15099*, 2023.

[23] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[24] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, 2014.

[25] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[26] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, 2023.

[27] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[28] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[29] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015.

[30] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[31] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, 2022.

[32] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[33] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

[34] Shiwei Li, Xiandi Luo, Haozhao Wang, Xing Tang, Ziqiang Cui, Dugang Liu, Yuhua Li, Xiuqiang He, and Ruixuan Li. Beyond higher rank: Token-wise input-output projections for efficient low-rank adaptation. *arXiv preprint arXiv:2510.23123*, 2025.

[35] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

[36] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*, 2017.

[37] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114, 2024.

[38] 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 *Forty-first International Conference on Machine Learning*, 2024.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[40] Zefang Liu and Jiahua Luo. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. *arXiv preprint arXiv:2405.00361*, 2024.

[41] Jian Ma, Xinchen Lyu, Jun Jiang, Longhao Zou, Chenshan Ren, Qimei Cui, and Xiaofeng Tao. Fouriercompress: Layer-aware spectral activation compression for efficient and accurate collaborative llm inference. *arXiv preprint arXiv:2510.16418*, 2025.

[42] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

[43] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

[44] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, 2011.

[45] Lin Ning, Harsh Lara, Meiqi Guo, and Abhinav Rastogi. Mode: Effective multi-task parameter efficient fine-tuning with a mixture of dyadic experts. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8248–8261, 2025.

[46] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall Press, USA, 3rd edition, 2009.

[47] Junyoung Park, Soo Yong Kim, Sang Heon Lee, and Jeonghwan Lee. Lfma: Parameter-efficient fine-tuning via layerwise fourier masked adapter with top-k frequency selection. In Francisco Acosta, Simone Azeglio, Bahareh Tolooshams, Chase van de Geijn, Christian Shewmake, Sophia Sanborn, and Nina Miolane, editors, *Proceedings of the 4th (2025) and 3rd (2024) NeurIPS Workshops on Symmetry and Geometry in Neural Representations*, volume 282 of *Proceedings of Machine Learning Research*, pages 415–427. PMLR, 2026.

[48] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

[49] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[51] Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3052–3064, 2024.

[52] Subhro Roy and Dan Roth. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*, 2016.

[53] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[54] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, 2019.

[55] Chongjie Si, Zhiyi Shi, Shifan Zhang, Xiaokang Yang, Hanspeter Pfister, and Wei Shen. Unleashing the power of task-specific directions in parameter efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[56] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[57] Mengyang Sun, Yihao Wang, Tao Feng, Dan Zhang, Yifan Zhu, and Jie Tang. A stronger mixture of low-rank experts for fine-tuning foundation models. *arXiv preprint arXiv:2502.15828*, 2025.

[58] Alex Tamkin, Dan Jurafsky, and Noah Goodman. Language through a prism: A spectral approach for multiscale language representations. *Advances in Neural Information Processing Systems*, 33:5492–5504, 2020.

[59] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *arXiv preprint arXiv:2404.19245*, 2024.

[60] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[61] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, 2023.

[62] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355, 2018.

[63] Fan Wang, Juyong Jiang, Chansung Park, Sunghun Kim, and Jing Tang. Kasa: Knowledge-aware singular-value adaptation of large language models. *arXiv preprint arXiv:2412.06071*, 2024.

[64] Hanqing Wang, Zeguan Xiao, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. *arXiv preprint arXiv:2406.09044*, 2024.

[65] Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *arXiv preprint arXiv:2205.12410*, 2022.

[66] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[67] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1740–1749, 2020.

[68] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[69] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[70] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.

[71] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[72] Dacao Zhang, Kun Zhang, Shimao Chu, Le Wu, Xin Li, and Si Wei. More: A mixture of low-rank experts for adaptive multi-task learning. *arXiv preprint arXiv:2505.22694*, 2025.

[73] Qianru Zhang, Peng Yang, Honggang Wen, Xinzhu Li, Haixin Wang, Fang Sun, Zezheng Song, Zhichen Lai, Rui Ma, Ruihua Han, et al. Beyond the time domain: Recent advances on frequency transforms in time series analysis. *arXiv preprint arXiv:2504.07099*, 2025.

[74] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

[75] Yibo Zhong, Haoxiang Jiang, Lincan Li, Ryumei Nakada, Tianci Liu, Linjun Zhang, Huaxiu Yao, and Haoyu Wang. Neat: Nonlinear parameter-efficient adaptation of pre-trained models. *arXiv preprint arXiv:2410.01870*, 2024.

[76] Yimeng Zhuang, Jing Zhang, and Mei Tu. Long-range sequence modeling with predictable sparse attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 234–243, 2022.

A Algorithm for FourierMoE

Algorithm 1 FourierMoE: Fourier Mixture-of-Experts Adaptation

Require: Pretrained weights $\mathbf{W}_0 \in \mathbb{R}^{M\times N}$, Training dataset $\mathcal{D}_{train}$, scaling factor η.
Require: Number of experts Z, active experts K, frequency count n, bandwidth $\mathcal{W}$.
Initialize Router: $\Phi \leftarrow$ random initialization.
Initialize Experts: For each expert $i \in \{1, \ldots, Z\}$:
  Sample active frequency indices $\Omega_i = \{(u, v)\}_1^n$ via Gaussian distribution $P_i(u, v)$ (Eq. 2).
  Initialize complex coefficients $\Theta_i = \{\mathbf{C}_i(u, v) \in \mathbb{C} \mid (u, v) \in \Omega_i\}$.
while not converged **do**
  Sample batch of inputs $x \sim \mathcal{D}_{train}$.
  1. Gating & Routing:
    Compute gating scores: $g = \text{Softmax}(\mathbf{W}_g x)$.
    Select Top-K experts: $\mathcal{S}(x) = \text{Top-}K(g)$.
  2. Spectral-to-Spatial Expert Construction:
    Initialize composite update $\Delta\mathbf{W} \leftarrow \mathbf{0}^{M\times N}$.
  for each selected expert $i \in \mathcal{S}(x)$ **do**
      Construct sparse spectral matrix $\mathbf{E}_i \in \mathbb{C}^{M\times N}$ from Θ_i.
      // Enforce Conjugate Symmetry (Theorem 2.1) for lossless real-valued adaptation
      $\mathbf{E}_i(-u, -v) \leftarrow \mathbf{E}_i^*(u, v), \quad \forall (u, v) \in \Omega_i$.
      Compute spatial expert update via IDFT:
      $\Delta\mathbf{W}_i \leftarrow \mathcal{F}^{-1}(\mathbf{E}_i)$ (Eq. 1).
      Accumulate: $\Delta\mathbf{W} \leftarrow \Delta\mathbf{W} + g_i \cdot \Delta\mathbf{W}_i$.
  end for
  3. Forward Pass:
    Compute output: $h = \mathbf{W}_0 x + \eta\Delta\mathbf{W}x$ (Eq. 14).
  4. Optimization:
    Calculate Loss: $\mathcal{L} = \mathcal{L}_{task}(h) + \lambda\mathcal{L}_{aux}(g)$ (Eq. 15 and Eq. 16).
    Update parameters $\{\Theta, \Phi\}$ via gradient descent.
end while
Ensure: Adapted parameters $\{\Theta, \Phi\}$.

B More Related Works

B.1 Parameter-efficient Fine-tuning

The escalating scale of LLMs renders FFT, a commonly used adaptation paradigm, computationally prohibitive [21]. PEFT addresses this bottleneck by freezing the pretrained backbone and updating only a minimal set of added or existing parameters [35]. For instance, LoRA [25] employs a pair of learnable low-rank matrices to approximate the learned weight updates. While effective for single-task adaptation, such PEFT methods that rely on a single tunable module often suffer from capacity constraints and task interference in multi-task scenarios due to a single parameter space shared across diverse or conflicting objectives [33, 59]. To alleviate these limitations, recent works incorporate the MoE architecture into PEFT, giving rise to MoPE methods [11, 57, 65, 14, 70]. These approaches employ a router to dynamically select lightweight expert modules conditioned on input tokens [3], enabling a more flexible allocation of task-specific capacity and potentially reducing task interference. However, existing MoPE methods primarily operate in the spatial domain where experts are constructed as parallel instantiations of lightweight PEFT modules. Such designs lack explicit mechanisms to encourage orthogonality or diversity among experts, which can lead to structural redundancy [15, 59]. Moreover, maintaining multiple spatial experts incurs additional parameter overhead, which partially compromises the fundamental efficiency goals of PEFT. Unlike prior MoPE methods, FourierMoE utilizes a router to dispatch tokens to experts specialized in distinct frequency bands, with each expert learning conjugate-symmetric complex coefficients in the Fourier domain.

B.2 Fourier Transform

Fourier transform has provided a powerful tool for analyzing global correlations and structural patterns in the frequency domain [73, 12]. It has been widely explored and used in various domains, such as optimizing long-sequence modeling [76, 20, 22], compressing activations [41], and improving the efficiency of the Transformer architecture [31]. Fourier transform and spectral adaptation have recently attracted increasing attention in the PEFT regime. FourierFT [16] pioneers this line of research by parameterizing weight updates in the spectral domain. It learns a set of coefficients within a sparse spectral matrix and reconstructs it via IDFT to obtain the weight updates. While achieving performance competitive with LoRA, FourierFT treats spectral components uniformly and does not explicitly account for the varying importance of different frequency bands, which may lead to suboptimal parameter allocation [47]. To refine this, LFMA [47] optimizes only the top-k coefficient coordinates with the largest magnitudes. However, these methods rely on static coefficient locations that cannot adapt to the dynamic evolution of gradients during fine-tuning or the distinct frequency signatures inherent in input tokens. Furthermore, they truncate the imaginary components of the complex spectrum, restricting the representation power of the learned updates. In contrast to prior works, FourierMoE integrates a frequency-adaptive router with specialized experts, enabling the composition of different frequency-specialized updates. Furthermore, by enforcing complex coefficient learning and conjugate symmetry, FourierMoE bridges the gap between spectral-domain efficiency and spatial-domain expressivity, achieving superior performance while maintaining high efficiency.

C Baselines

To evaluate our method's effectiveness, we compare FourierMoE against FFT, PEFT, and MoPE baselines, including:
FFT initializes the foundation model with its pretrained weights and updates all parameters.
FFT MoE fully fine-tunes the upcycled MoE, which transforms the pretrained dense model into a MoE architecture by duplicating pretrained weights to create multiple experts and integrating a learnable router.
LoRA [25] reparameterizes the weight updates into a pair of tunable low-rank matrices, which can be merged with the foundation model after fine-tuning.
rsLoRA [28] analyzes the scaling factor of LoRA and proposes to stabilize LoRA's learning trajectory with a factor of $\frac{\alpha}{\sqrt{r}}$ instead of $\frac{\alpha}{r}$.
DoRA [38] factorizes pretrained weights into magnitude and direction, updating only the direction via low-rank adaptation while learning the magnitude separately.
PiSSA [42] performs singular value decomposition (SVD) on pretrained weights to obtain a principled low-rank initialization for fine-tuning.
MiLoRA [64] also leverages SVD on pretrained weights, but freezes the principal components and fine-tunes only the minor low-rank components.
KaSA [63] applies SVD to the pretrained weights, truncates the minor components, and subsequently learns the weight updates in the SVD space.
LoRA-Dash [55] first runs a short LoRA warm-up to identify task-specific directions by projecting the learned updates onto the singular directions of pretrained weights, and then jointly learns the corresponding coordinate scalars together with the low-rank adapters during fine-tuning.
NEAT [75] introduces a lightweight network that takes the pretrained model weights as input and learns a nonlinear transformation to approximate the weight updates during fine-tuning.
TopLoRA [34] dynamically adjusts the LoRA weights by using a token-wise diagonal matrix generated by an additional, tunable projector network.
MELoRA [51] concatenates multiple mini LoRA modules in parallel along the diagonal to construct a block diagonal LoRA matrix, achieving higher effective rank without additional parameter overhead.
LFMA [47] selects and learns the Top-K high-magnitude Fourier coefficients in the spectral domain, then converts them back to the spatial domain via the IDFT to produce model weight updates.
FourierFT [16] learns the Fourier coefficients in the spectral domain, then converts them back to the spatial domain via the IDFT to produce model weight updates.
MoLoRA [70] integrates the MoE framework with PEFT by replacing the experts (usually initialized as the feed-forward network copies) with lightweight LoRA modules, which are selectively activated for each token via a routing mechanism.

AdaMoLE [40] replaces the static top-k selection in the LoRA-MoE framework with dynamic thresholding mechanism to adaptively activate the most relevant experts based on input context.
HydraLoRA [59] introduces an asymmetric LoRA-MoE framework, characterized by a shared $\mathbf{A}$ matrix to capture general knowledge and multiple task-specific matrices $\mathbf{B}$ that are dynamically combined through a routing mechanism.
GOAT [11] proposes a SVD-structured MoE framework that adaptively activates the experts initialized with distinct singular-value segments derived from the pretrained weights. In addition, it employs a theoretical scaling scheme to align the low-rank adaptation trajectories with those of fully fine-tuned MoE.

D Details of Datasets and Benchmarks

D.1 Commonsense Reasoning Datasets

For the commonsense reasoning tasks, we fine-tune LLMs on the Commonsense170K dataset [26]. It contains data samples from eight datasets: ARC-e, ARC-c [8], OBQA [43], SIQA [54], WinoGrande [53], HellaSwag [71], BoolQ [7], and PIQA [1]. We evaluate the fine-tuned models on the test splits of each dataset and report accuracy. The statistics of the datasets are presented in Table 9.

Table 9: Statistics of the sub-datasets comprising the Commonsense170K dataset used for multi-task fine-tuning.

Dataset	Domain	# Train	# Test	Task Type	Answer
ARC-e	Natural Science	2.3K	2.3K	Question Answering	Option
ARC-c	Natural Science	1.1K	1.1K	Question Answering	Option
BoolQ	Wikipedia	9.4K	3.2K	Text Classification	Yes/No
OpenBookQA (OBQA)	Science Facts	5.0K	500	Question Answering	Option
PIQA	Physical Interaction	16.1K	1.8K	Question Answering	Option
SIQA	Social Interaction	33.4K	1.9K	Question Answering	Option
HellaSwag	Video Caption	39.9K	10K	Sentence Completion	Option
WinoGrande	Winograd Schemas	63.2K	1.2K	Fill in the Blank	Option

D.2 Math Reasoning Datasets

We adopt the Math10K dataset [26] to adapt LLMs for math reasoning tasks. This dataset consists of training data from AQuA [36] and GSM8K [9]. In addition, we evaluate the fine-tuned models on six math benchmarks: GSM8K [9], SVAMP [48], MultiArith [52], AddSub [24], AQuA [36], and SingleEQ [29]. We report accuracy for all benchmarks. The statistics of the datasets and benchmarks are shown in Table 10.

Table 10: Statistics of sub-datasets comprising the Math10K dataset used for multi-task fine-tuning. In the Math10K dataset, only GSM8K and AQuA provide training samples. The training sample counts of other datasets are marked with the "-" symbol, indicating that we do not use their training data for fine-tuning.

Dataset	Domain	# Train	# Test	Task Type	Answer
AddSub	Arithmetic Word Problems	-	395	Math Word Problem Solving	Number
AQuA	GRE/GMAT Math	100K	254	Multiple-Choice Question Answering	Option
GSM8K	Grade School Math Word Problems	8.8K	1.3K	Math Word Problem Solving	Number
MultiArith	Multi-step Math Word Problems	-	600	Math Word Problem Solving	Number
SingleEQ	Grade School Algebra Word Problems	-	508	Math Word Problem Solving	Number
SVAMP	Arithmetic Word Problems	-	1K	Math Word Problem Solving	Number

D.3 GLUE Dataset

For NLU tasks, we fine-tune RoBERTa-large on seven datasets from GLUE [62], including CoLA, SST-2, MRPC, QQP, MNLI, QNLI, and RTE. We evaluate the fine-tuned model on the test splits of each dataset and report accuracy. Dataset statistics are shown in Table 11.

Table 11: Statistics of the seven GLUE datasets for single-task fine-tuning.

Dataset	Domain	# Train	# Test	Task Type	Answer
CoLA	Miscellaneous	8.5K	1.0K	Acceptability	Label Text
SST-2	Movie Reviews	67.3K	1.8K	Sentiment Analysis	Label Text
MRPC	News	3.6K	1.7K	Paraphrase	Label Text
QQP	Social QA	364K	391K	Paraphrase	Label Text
MNLI	Miscellaneous	393K	19.6K	Natural Language Inference	Label Text
QNLI	Wikipedia	105K	5.4K	Natural Language Inference	Label Text
RTE	News & Wikipedia	2.4K	3K	Natural Language Inference	Label Text

D.4 Image Classification Datasets

For image classification tasks, we fine-tune CLIP ViT-B/32 on seven datasets, including Cars [30], DTD [6], EuroSAT [23], GTSRB [56], RESISC45 [5], SUN397 [66], and SVHN [44]. We evaluate the fine-tuned model on the test splits of each dataset and report accuracy. The statistics of datasets and benchmarks are shown in Table 12.

Table 12: Statistics of the seven image classification datasets for single-task fine-tuning.

Dataset	Domain	# Train	# Test	Task Type	Answer
StanfordCars (Cars)	Automobiles	8.1K	8.0K	Image Classification	Class Label
DTD	Textures	3.7K	1.8K	Image Classification	Class Label
EuroSAT	Satellite	21.6K	2.7K	Image Classification	Class Label
GTSRB	Traffic Signs	26.6K	12.6K	Image Classification	Class Label
RESISC45	Aerial Imagery	18.9K	6.3K	Image Classification	Class Label
SUN397	Scenes	76.1K	21.7K	Image Classification	Class Label
SVHN	Street Digits	73.2K	26.0K	Image Classification	Class Label

E Training Details

We follow the prior work [33] to adopt a multi-task training setup for commonsense and math reasoning tasks, where models are trained on a mixed dataset and evaluated on different benchmarks. In addition, we employ a single-task training setup for image classification and NLU tasks, in which models are trained and evaluated separately on each task. To ensure optimal performance, we meticulously tune the hyperparameters, including the learning rate, batch size, number of epochs, scaling value η, and the load-balancing loss weight λ, and other hyperparameters. The full hyperparameter configurations for all experimental settings are provided: commonsense reasoning in Table 13, math reasoning in Table 14, image classification in Table 15, and NLU in Table 16.

Table 13: Detailed configurations used for fine-tuning LLaMA-2 7B and Gemma 7B models on commonsense reasoning tasks.

Hyperparameters	Commonsense Reasoning	
	LLaMA-2 7B	Gemma 7B
Optimizer	AdamW	
LR	8e-2	1.5e-1
LR Scheduler	Linear	
Max Seq. Len.	256	
Batch Size	16	
Accumulation Steps	16	
Dropout	0.05	
Warmup Ratio	0.03	0.05
# Epochs	1	
Spectral Coefficients n	8192	
Placement	Q,K,V,Up,Down,Gate	
Scaling Factor η	96	128
Load-balancing Scaling λ	0.002	0.001
# Experts	8	
Top-K	4	

Table 14: Detailed configurations used for fine-tuning LLaMA-3 8B and Qwen2.5-14B models on math reasoning tasks.

Hyperparameters	Math Reasoning	
	LLaMA-3 8B	Qwen2.5-14B
Optimizer	AdamW	
LR	2e-4	1e-4
LR Scheduler	Linear	
Max Seq. Len.	1024	
Batch Size	8	
Accumulation Steps	16	
Dropout	0.05	
Warmup Ratio	0.05	
# Epochs	1	
Spectral Coefficients n	8192	
Placement	Q, K, V, O, Up, Down, Gate	
Scaling Factor η	128	2
Load-balancing Scaling λ	0.001	0.002
# Experts	8	
Top-K	2	

Table 15: Detailed configurations used for fine-tuning CLIP ViT-B/32 on image classification tasks.

Hyperparameters	Cars	DTD	EuroSAT	GTSRB	RESISC45	SUN397	SVHN
Optimizer	AdamW						
Expert LR	8e-5	1e-2	5e-2	2e-1	5e-2	5e-2	1.5e-1
Router LR	5e-5	1e-4	2e-4	2e-4	2e-4	2e-4	2e-4
Head LR	3e-4	1e-3	1e-3	1e-3	5e-3	1e-3	1e-3
LR Scheduler	Cosine						
Max Seq. Len.	50						
Batch Size	512	128	512	512	512	512	512
Accumulation Steps	1						
Dropout	0.0						
Warmup Ratio	0.03	0.03	0.05	0.05	0.03	0.05	0.05
# Epochs	280	100	30	50	45	20	20
Spectral Coefficients n	3008						
Placement	Q,V						
Scaling Factor η	192	32	32	256	32	32	192
Load-balancing Scaling λ	1e-3						
# Experts	8						
Top-K	2						

F Additional Experimental Results

F.1 Effect of Frequency Bandwidth $\mathcal{W}$ on Expert Specialization

As defined in Eq. 2, each expert in FourierMoE is assigned within a specific frequency bandwidth $\mathcal{W}$, which controls the range of spectral components it can access and thus governs the degree of expert specialization. To explain the non-monotonic performance trend observed in Section 3.3, we analyze how varying $\mathcal{W}$ shapes the spectral organization of experts, as visualized in Figure 6. When $W = 0.06$, experts are restricted to narrow frequency bands around their assigned centers, limiting per-expert spectral coverage and expressiveness. Increasing $\mathcal{W}$ to 0.12 expands frequency coverage while maintaining well-separated annular structures across experts. This regime strikes a favorable balance between expressiveness and specialization, enabling experts to capture richer spectral patterns without sacrificing functional diversity, which aligns with the performance reported in Table 5. In contrast, further increasing the bandwidth to 0.24 and beyond introduces spectral overlap across experts, weakening specialization and reducing parameter efficiency. In the extreme regime ($\mathcal{W} \geq 0.96$), expert spectra become highly overlapping and near-uniform, collapsing the MoE into a collection of functionally similar modules. Such loss of specialization can lead to ambiguous routing decisions and unstable optimization, ultimately degrading model performance. These observations reveal a trade-off in FourierMoE: while increasing $\mathcal{W}$ enhances per-expert expressiveness, excessively wide bandwidths undermine cross-expert spectral specialization.

F.2 Expert Similarity

To better analyze expert similarity, we reconstruct the Fourier experts into the spatial domain via IDFT and compute pairwise cosine similarity among the expert weights of CLIP ViT-B/32 fine-tuned on EuroSAT. As shown in Figure 7, the absolute cosine similarity between any pair of distinct experts

Table 16: Detailed configurations used for fine-tuning RoBERTa-large on NLU tasks.

Hyperparameters	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE
Optimizer				AdamW			
Expert LR	5e-2	1.2e-1	9e-2	1e-1	1e-5	9e-2	1e-1
Router LR	1.5e-4	1e-4	2e-4	3.9e-4	2e-4	1.5e-4	1.5e-4
Head LR	1e-2	5e-4	1e-3	8.4e-4	5e-3	1e-3	6.5e-3
LR Scheduler				Linear			
Max Seq. Len.	256	128	512	128	512	512	512
Batch Size				32			
Accumulation Steps				1			
Dropout				0.0			
Warmup Ratio				0.06			
# Epochs	80	10	30	15	10	30	60
Spectral Coefficients n				1008			
Placement				Q,V			
Scaling Factor η	115	138	160	64	96	72	136
Load-balancing Scaling λ	1.5e-3	1.5e-3	2e-3	2.5e-3	3.5e-3	2.5e-3	1.7e-3
# Experts				8			
Top-K				2			

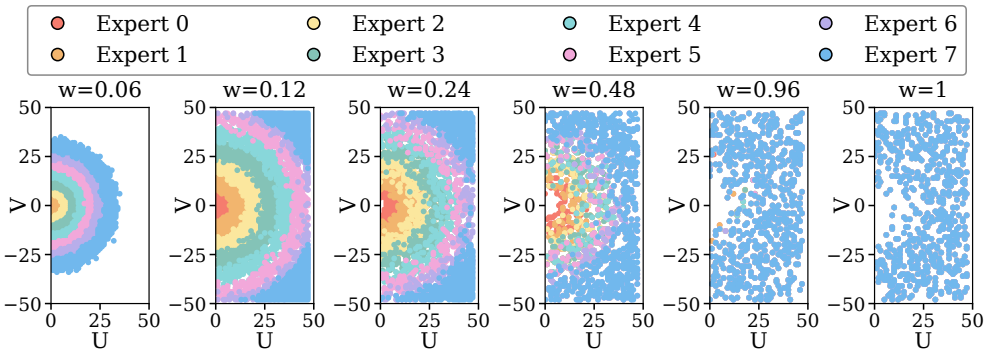

Figure 6: Distributions of expert spectral coefficients under different frequency bandwidths $\mathcal{W}$. Moderate bandwidths expand per-expert frequency coverage while preserving well-separated spectral structures, whereas overly large $\mathcal{W}$ induces substantial overlap, leading to degraded specialization.

is below 5×10^{-4}. Specifically, 27 of the 28 distinct pairs have magnitudes below 2×10^{-9}, while the largest magnitude occurs between Experts 1 and 2, with a similarity of -4.917×10^{-4}. These results indicate negligible redundancy among experts.

G Complexity Analysis

We analyze the computational and memory complexity of FourierMoE compared to standard FFT and LoRA fine-tuning. Let M, N denote the weight matrix dimensions, Z the total number of experts, K the number of active experts, and n the number of active spectral coefficients per expert (where $n \ll MN$).

Parameter Efficiency. Standard FFT requires updating $\mathcal{O}(MN)$ parameters per layer. LoRA reduces this to $\mathcal{O}(r(M+N))$ where r is the rank of the lightweight tunable matrices. FourierMoE achieves sparsity by learning only the active spectral coefficients and the router. The total trainable parameter count is:

$$\mathcal{P}_{\text{FourierMoE}} = \mathcal{O}(Z \cdot n + Z \cdot N). \tag{17}$$

Since the crucial adaptation information is concentrated in a small subset of dominant frequency components [47, 16], we can set n such that $Z \cdot n \ll r(M+N)$, allowing us to scale the number of experts Z without incurring significant memory overhead.

Inference Latency. The computational cost of FourierMoE during inference consists of the routing operation and the expert reconstruction.

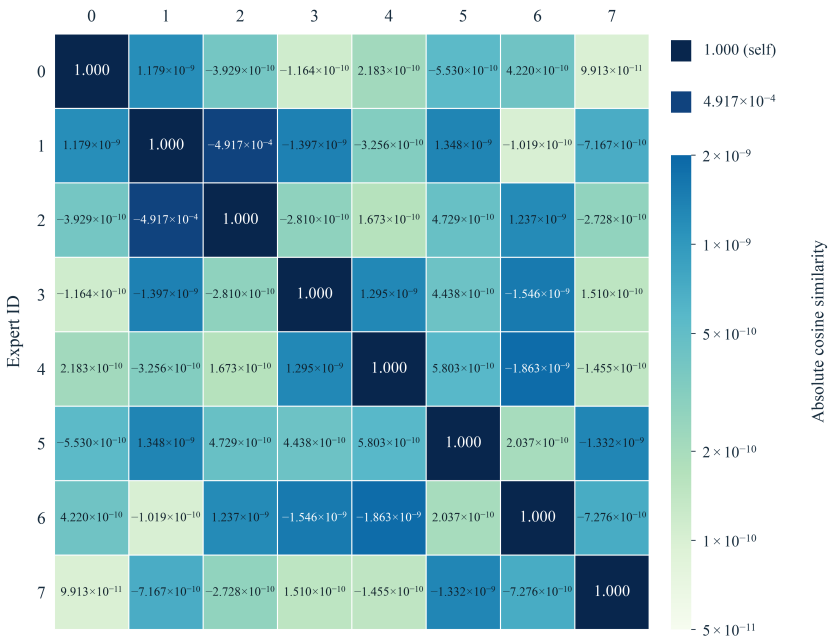

Figure 7: Pairwise cosine similarity among the expert weights of CLIP ViT-B/32 fine-tuned with FourierMoE on EuroSAT. Numbers denote cosine similarities, while colors represent absolute values.

- **Routing Overhead:** The gating mechanism requires a lightweight projection $\mathcal{O}(N \cdot Z)$, which is negligible compared to the forward pass of the LLM layers.
- **Reconstruction Cost:** Constructing the update $\Delta\mathbf{W}$ requires summing K experts. Utilizing the Fast Fourier Transform, the reconstruction complexity is $\mathcal{O}(K \cdot MN \log(MN))$.

Crucially, the inference cost depends only on the active set size K, not the total expert pool Z. This allows FourierMoE to increase model capacity (via larger Z) while maintaining constant-time inference complexity, satisfying the condition $\mathcal{O}(K) \approx$ const with respect to Z. Furthermore, by caching the reconstructed $\Delta\mathbf{W}$ for static tasks or utilizing parallelized GPU-optimized FFT kernels, the latency overhead remains within a tractable margin for real-time deployment.

H Spectral-Spatial Rank Duality

In this section, we establish a rigorous theoretical connection between the spectral composition of our experts and the algebraic properties of the resulting weight updates in the spatial domain. Specifically, we investigate the relationship between the spectral sparsity of an expert $\mathbf{E}_i$ and the matrix rank of its spatial counterpart $\Delta\mathbf{W}_i$.

We demonstrate that FourierMoE generalizes low-rank adaptation methods (e.g., LoRA) by enabling dynamic rank allocation. Unlike static low-rank methods that fix the rank r globally, FourierMoE allows the router to dynamically assign a specific capacity to each input by selecting experts with varying frequency bands.

H.1 Rank Properties of the Fourier Basis

To analyze the rank of the model's weight update $\Delta\mathbf{W}$, we examine the algebraic structure of the Fourier basis kernel $\mathbf{B}_{u,v}$ defined in Eq. (1).

Lemma H.1 (Rank-1 Property of Fourier Kernels). *For any frequency coordinate (u, v), the Fourier basis kernel $\mathbf{B}_{u,v} \in \mathbb{C}^{M\times N}$ is a rank-1 matrix.*

Proof. Recall the definition of the kernel entry at spatial index (q, y):

$$\mathbf{B}_{u,v}(q, y) = \exp\left(j2\pi\frac{uq}{M}\right) \cdot \exp\left(j2\pi\frac{vy}{N}\right). \tag{18}$$

Let $\mathbf{f}_u \in \mathbb{C}^M$ and $\mathbf{g}_v \in \mathbb{C}^N$ be vectors such that the entries are defined as $\mathbf{f}_u(x) = e^{j2\pi uq/M}$ and $\mathbf{g}_v(y) = e^{j2\pi vy/N}$. We can express the matrix $\mathbf{B}_{u,v}$ as the outer product:

$$\mathbf{B}_{u,v} = \mathbf{f}_u \mathbf{g}_v^T. \tag{19}$$

Since $\mathbf{B}_{u,v}$ is formed by the outer product of two non-zero vectors, it possesses exactly one non-zero singular value. Thus, $\mathrm{rank}(\mathbf{B}_{u,v}) = 1$. □

H.2 Spectral Sparsity Bounds Spatial Rank

Building on Lemma H.1, we derive the upper bound on the rank of the spatial update generated by an expert defined by a sparse set of active frequencies.

Theorem H.2 (**Spectral Sparsity-Rank Inequality**). *Let $\mathbf{E}_i$ be a frequency-specialized expert with a set of active frequencies Ω_i satisfying the conjugate symmetry condition (Theorem 2.1). Let $K = |\Omega_i|$ be the spectral sparsity (the total number of non-zero frequency coefficients). The rank of the resulting spatial update $\Delta\mathbf{W}_i$ is bounded by:*

$$\mathrm{rank}(\Delta\mathbf{W}_i) \leq \min(M, N, K). \tag{20}$$

Proof. From Eq. (1), the spatial update is a linear combination of basis kernels:

$$\Delta\mathbf{W}_i = \frac{1}{MN} \sum_{(u,v)\in\Omega_i} \mathbf{F}(u,v) \cdot \mathbf{B}_{u,v}. \tag{21}$$

Using the subadditivity property of matrix rank (the rank of a sum is less than or equal to the sum of the ranks), we have:

$$\begin{aligned} \mathrm{rank}(\Delta\mathbf{W}_i) &= \mathrm{rank}\left(\sum_{(u,v)\in\Omega_i} c_{u,v}\mathbf{B}_{u,v} \right) \\ &\leq \sum_{(u,v)\in\Omega_i} \mathrm{rank}(c_{u,v}\mathbf{B}_{u,v}). \end{aligned} \tag{22}$$

By Lemma H.1, $\mathrm{rank}(\mathbf{B}_{u,v}) = 1$. Therefore:

$$\mathrm{rank}(\Delta\mathbf{W}_i) \leq \sum_{(u,v)\in\Omega_i} 1 = |\Omega_i| = K. \tag{23}$$

Since the rank of any matrix in $\mathbb{R}^{M\times N}$ cannot exceed its dimensions, it holds that $\mathrm{rank}(\Delta\mathbf{W}_i) \leq \min(M, N, K)$. □

Remark on Real-Valued Updates. Due to the Conjugate Symmetry Condition (Theorem 2.1), active frequencies must appear in pairs $((u,v), (\langle -u\rangle_M, \langle -v\rangle_N))$ to ensure $\Delta\mathbf{W}_i \in \mathbb{R}^{M\times N}$. A symmetric pair combines to form a real-valued sinusoidal wave:

$$c\mathbf{B}_{u,v} + c^*\mathbf{B}_{\langle -u\rangle_M, \langle -v\rangle_N} = 2|c| \cos\left(2\pi\left(\frac{uq}{M} + \frac{vy}{N}\right) + \angle c\right). \tag{24}$$

While this sum is real-valued, it is formed by the sum of two rank-1 complex matrices. Thus, each symmetric pair contributes at most 2 to the rank of the spatial matrix.

H.3 Input-Dependent Dynamic Rank Allocation

The insights from Theorem H.2 reveal the fundamental mechanism of FourierMoE. By initializing experts with Gaussian filters of varying bandwidths $\mathcal{W}_i$ (Eq. 2), we effectively create a bank of experts with different rank capacities.

Corollary H.3 (**Router as a Rank Selector**). *Let the router $\mathcal{G}(x)$ select a subset of experts $\mathcal{S}(x)$. The effective rank of the update applied to input x is:*

$$r_{eff}(x) = \mathrm{rank}\left(\sum_{i\in\mathcal{S}(x)} \mathcal{G}(x)_i \Delta\mathbf{W}_i \right) \leq \sum_{i\in\mathcal{S}(x)} |\Omega_i|. \tag{25}$$

This leads to a critical theoretical distinction between FourierMoE and standard LoRA:

- **LoRA (Static Rank):** Forces a fixed rank r for all inputs, i.e., $\Delta\mathbf{W} = \mathbf{BA}$. The expressivity is static regardless of input complexity.
- **FourierMoE (Dynamic Rank):** The router dynamically determines the required complexity. For "easy" tokens (e.g., stop words), the router may select Low-Bandwidth Experts (small $|\Omega_i|$), applying a *Low-Rank* update. For "hard" tokens (e.g., reasoning steps), it may select High-Bandwidth Experts (large $|\Omega_i|$), applying a *High-Rank* update.

H.4 Orthogonality and Global Receptive Field

Finally, we address why spectral rank-1 updates differ from spatial rank-1 updates. In standard low-rank adaptation, the basis vectors (columns of $\mathbf{A}$ and $\mathbf{B}$) are often learned to be sparse or localized in the spatial domain. In contrast, the Fourier basis vectors $\mathbf{f}_u$ and $\mathbf{g}_v$ are *dense* in the spatial domain; every element has magnitude $1/\sqrt{M}$.

Proposition H.4 (**Global Information Flow**). *A rank-1 update in the Fourier domain effects a global update in the spatial domain. Specifically, modifying a single spectral coefficient $\mathbf{F}(u, v)$ updates every weight in $\Delta\mathbf{W}$ simultaneously with constant magnitude, merely shifting the phase.*

This property implies that our FourierMoE is exceptionally efficient at learning *global* features (via low frequencies) and *distributed* patterns, whereas standard spatial sparsity methods struggle to propagate information globally without increasing parameter density. The Fourier transform acts as a maximally incoherent basis change, allowing sparse spectral experts to model dense spatial correlations efficiently.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper provides detailed experimental results and related theory to accurately reflect the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: The paper does not discuss the limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The full set of assumptions and complete proofs are provided in Section 2 and Appendix H.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details are available in Appendix A and E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code is still being organized. We plan to release it publicly once it is ready.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all experimental details in Section 3, Appendix D, and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars or statistical significance tests due to the large computational cost of repeated runs across multiple model families and benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on the computer resources is provided in Section 3 and Table 7.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not discuss the potential positive societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all models and datasets used in the paper and follow their license terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. LLMs are only used to assist with grammar checking during the paper writing process and do not impact the core methodology, scientific rigor, or originality of our research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.