Frame2Freq: Spectral Adapters for Fine-Grained Video Understanding

Supplementary Material

A. Implementation Details

A.1. Frequency Spectral Analysis

Video Spectral Analysis. To visualize and compare the motion characteristics of different action classes, we compute full 3D Fourier spectra directly from raw videos. Each video is first uniformly sampled to 16 grayscale frames and resized to 224×224 pixels. This produces a spatiotemporal volume $V \in \mathbb{R}^{H\times W\times T}$ with H=W=224 and T=16. A 3D FFT is then applied to the entire tensor, $F = \text{FFT}_{(x,y,t)}(V)$, followed by a centered frequency shift. The magnitude (or power) spectrum is log-compressed to stabilize large dynamic ranges. To emphasize discriminative patterns, we optionally remove the DC component and use spectral whitening (i.e., division by the radial frequency radius). We take the central temporal slice of the 3D spectrum, which gives a stable 2D frequency map that captures spatial–temporal motion patterns without mixing across temporal bands. Finally, we calculate the class-wise mean spectra to verify if the patterns stay relevant across multiple samples in the dataset. We perform this by averaging FFT magnitudes across all videos in a class after normalizing their shapes and exporting them as high-resolution maps.

Frequency Discriminability Analysis. To quantify how informative each frequency is for distinguishing actions, we perform a Frequency Discriminability Analysis inspired by Analysis of Variance (ANOVA) [11]. Temporal embeddings $X_i \in \mathbb{R}^{T\times D}$ extracted from the backbone (after adapter insertion) are transformed using a 1D FFT along time, giving $\tilde{X}i(f)$. This operation decomposes each embedding dimension into frequency components, capturing periodicity, phase changes, and motion rhythms encoded by the model. We convert these into scalar frequency responses by computing the mean spectral power across embedding dimensions, $P_i(f) = \mathbb{E}_d|\tilde{X}_i(f,d)|^2$, resulting in a power vector for each clip that summarizes its activity across frequencies. Clips are then grouped according to their ground-truth class labels, allowing us to compute class-wise means and within-class variances at each frequency. Following the ANOVA principle, the discriminability score is obtained by taking the ratio between the between-class and within-class variance at each temporal frequency, $D(f) = \frac{\text{Between}(f)}{\text{Within}(f)+\epsilon}$,where a small ϵ ensures numerical stability. Finally, the discriminability curve is normalized across frequencies to produce $\hat{D}(f)$, which represents a probability-like distribution indicating the amount of discriminative power present at each temporal band. Algorithm 1 provides an overview of the implementation of our Frequency Discriminability Analysis.

Algorithm 1: Frequency Discriminability Analysis

Input: Temporal embeddings $\{X_i \in \mathbb{R}^{T\times D}\}_{i=1}^N$, class labels $\{y_i\}_{i=1}^N$

Output: Normalized discriminability scores $\{\hat{D}(f)\}_{f=1}^F$

Step 1: Compute spectral power per clip

foreach *clip* i **do**

$\tilde{X}_i(f) \leftarrow \text{FFT}_t(X_i)$; `// 1D FFT along temporal axis`

$P_i(f) \leftarrow \mathbb{E}_d\,|\tilde{X}_i(f,d)|^2$; `// Spectral power (mean over D)`

Step 2: Group by class labels

Let $\mathcal{C}$ be the set of unique classes.

For each $c \in \mathcal{C}$, define $\mathcal{P}_c(f) = \{P_i(f) \mid y_i = c\}$.

Step 3: Compute per-frequency discriminability (ANOVA principle)

for $f = 1$ **to** F **do**

$\mu(f) \leftarrow \frac{1}{N}\sum_i P_i(f)$; `// Global mean`

for $c \in \mathcal{C}$ **do**

$\mu_c(f) \leftarrow \frac{1}{|\mathcal{P}_c|}\sum_{p\in\mathcal{P}_c(f)} p$; `// Class-wise mean`

$\text{Between}(f) \leftarrow \sum_{c\in\mathcal{C}} |\mathcal{P}_c|\,[\mu_c(f) - \mu(f)]^2$

$\text{Within}(f) \leftarrow \sum_{c\in\mathcal{C}}\sum_{p\in\mathcal{P}_c(f)}[p - \mu_c(f)]^2$

$D(f) \leftarrow \frac{\text{Between}(f)}{\text{Within}(f)+\epsilon}$; `// Frequency-wise discriminability`

Step 4: Normalize across frequencies

$\hat{D}(f) \leftarrow \frac{D(f)}{\sum_{f'=1}^F D(f')}$; `// Normalize such that` $\sum_f \hat{D}(f) = 1$

return $\{\hat{D}(f)\}_{f=1}^F$

A.2. Model Implementation

All our models are trained using frozen CLIP [37] or DINOv2 [31] backbones, updating only the Frame2Freq adapter and the linear classification head. Training is performed on 16x NVIDIA A100 GPUs for SSv2 [12] and Diving48 [21], whereas 4x NVIDIA A100 GPUs were used for training the fine-grained human-object interaction datasets (Drive&Act [29], IKEA-ASM [3], and HRI-30 [14]) with DDP used for GPU-parallelization. We train for 60 epochs with AdamW, cosine decay, batch size 32, learning rate 1e-3, weight decay 0.05, and spatial resolution 224×224. We also use 2 warmup epochs for all the datasets to ensure smooth learning of parameters. FFT operations are

computed using batched CUDA kernels without additional regularization or test-time augmentation. This setup ensures that all reported improvements arise solely from the frequency-aware adapter and its integration into the frozen spatial backbone, isolating the contribution of spectral modeling to fine-grained temporal understanding.

Implementation of Frequency Transforms. We apply STFT per spatial location and channel by reshaping features to $(B \cdot H \cdot W \cdot C_a, T)$ and using a Hann window with $n_{fft} = min(32, T/2)$ and $hop = n_{fft}/4$ ($\sim 75\%$ overlap) and returns complex tensors. Complex spectra are processed by concatenating real/imaginary parts (phase preserved), refined via depthwise Conv3D in the time–frequency grid, and reconstructed with iSTFT using identical parameters.

We now describe the dataset-specific preprocessing and sampling strategies used across all benchmarks.

Diving48. Diving48 [21] contains fast, tightly synchronized diving sequences where frame-to-frame changes are minimal, so fine-grained temporal cues are crucial. We uniformly sample 32 frames per clip and apply RandAugment with random resized cropping and horizontal flips. Frame2Freq is inserted after the MHSA layer in every transformer block. At inference, we use a single temporal view with three spatial crops, matching standard PEFT evaluation practice.

Something Somethingv2. Something-Something V2 [12] is built around short, object-centric interactions where motion speed varies sharply across classes. We uniformly sample 16 or 32 frames and apply the standard SSv2 augmentations: RandAugment, random resized crops, and color jitter. Following prior PEFT setups, we use two Frame2Freq adapters per block for ViT-B/16 and a single adapter for ViT-L/14. In inference, we use one temporal clip and three spatial crops to match established evaluation practice.

Drive&Act. Drive&Act [29] features in-cabin driver monitoring with fine-grained, nearly symmetric actions such as reaching, picking, or placing objects. We follow the standard Drive&Act preprocessing pipeline: sampling 16 frames with a frame interval of 2, and applying a sequence of augmentations, including resizing, Random Resized Crop, and horizontal flips. We insert a single Frame2Freq adapter per transformer block, as this lightweight variant avoids overfitting on the smaller domain. During inference, we use three temporal clips and a spatial center crop to maintain consistency with prior work.

IKEA-ASM.IKEA-ASM [3] contains long-horizon assembly sequences with frequent nearly symmetric motions, such as pick up vs. lay down components or tightening vs. loosening screws. We sample 16 frames per clip at a frame interval of 2 and apply the standard augmentation pipeline used in prior assembly-action work, which includes resizing, Random Resized Crop, and flips. Similar to Drive&Act, we use one adapter per block and utilize three temporal clips per video during inference.

HRI-30.HRI-30 [14] captures close-range human–robot collaborative tasks with subtle object exchanges, handovers, and nearly symmetric motion phases that demand precise temporal cues. We uniformly sample 16 frames per clip and follow the same preprocessing protocol as Drive&Act and IKEA-ASM: RandomResizedCrop, horizontal flips, and per-frame normalization. Given the dataset's small size and fine-grained temporal structure, we again employ one adapter per block and one temporal clip and three spatial crops to match established evaluation practice.

B. Additional Experiments and Analysis

B.1. Variant Selection Guide

To better understand the behavior of the two variants, we analyze performance across action categories with different temporal characteristics (Table 7). On Diving48, actions involving a single characteristic temporal scale (e.g., no somersaults or twists) show comparable performance between ST and MS variants. In contrast, actions composed of multiple somersaults or twists exhibit *overlapping temporal frequencies*, where Frame2Freq-MS provides clear gains. A similar pattern emerges on HRI-30. Frame2Freq-MS improves performance on Movement+Manipulation classes that involve multi-scale temporal dynamics (simultaneous body and hand motion), while Frame2Freq-ST performs competitively on Pure Movement and Pure Manipulation categories characterized by a dominant temporal scale. Overall, these findings indicate that the single-scale variant is well suited for datasets dominated by a single characteristic temporal frequency, whereas the multi-scale variant is advantageous for complex actions with overlapping temporal structures.

Method	HRI-30			Diving48		
	Pure Movement	Pure Manip.	Movement+ Manip.	Single Scale	Multiple Scale	Other Classes
F2F-ST	97.9	95.4	69.1	89.9	65.9	73.9
F2F-MS	98.6	97.5	73.8	94.9	85.5	92.8

Table 7. Performance breakdown by temporal complexity across HRI-30 and Diving48, illustrating that Frame2Freq-ST excels in single-scale settings while MS variant benefits multi-scale actions.

B.2. Throughput Analysis

We compare the computational overhead of Frame2Freq-MS against existing Image-to-Video PEFT methods in Table 8. Despite integrating frequency transforms in adapters across all transformer layers, Frame2Freq-MS maintains competitive efficiency with 7.3M trainable parameters and 314 GFLOPs. The measured inference latency (13.11 ms) is comparable to prior PEFT approaches such as ST-Adapter (12.00 ms) and remains significantly more efficient than heavier designs like VitaCLIP and M2-CLIP. These results

confirm that the spectral adaptation introduces minimal runtime overhead while preserving strong empirical gains.

Method	Params	GFLOPs	Time (ms)
VitaCLIP [45]	29.0	1128	14.09
M2-CLIP [43]	14.8	421	13.62
ST-Adapter [32]	7.1	325	12.00
Frame2Freq-MS (Ours)	7.3	314	13.11

Table 8. Throughput Analysis of Frame2Freq in comparison to existing Image-to-Video PEFT methods.

Width	D&A	IKEA	SSv2	Diving48
96	81.27	74.26	–	–
192	**82.04**	**78.06**	68.9	91.2
384	79.63	75.87	**69.7**	**92.2**
768	–	–	69.5	91.1

Table 9. Impact of adapter width on performance of Frame2Freq adapter in 4 datasets.

B.3. Impact of Adapter Width

Table 9 reports the effect of varying the adapter bottleneck width across four datasets. A clear pattern emerges: smaller widths (192) perform best on fine-grained human–object datasets such as Drive&Act and IKEA-ASM, where motion signals are subtle and larger adapters tend to overfit. In contrast, higher-capacity adapters benefit motion-rich datasets like SSv2 and Diving48, where the 384-width variant captures more complex temporal variations and achieves the highest accuracy. These results highlight that frequency-sensitive temporal adaptation does not require uniformly large adapters; instead, optimal width depends on the underlying motion complexity and size of each dataset.

B.4. Frequency Analysis in Nearly Symmetric Actions

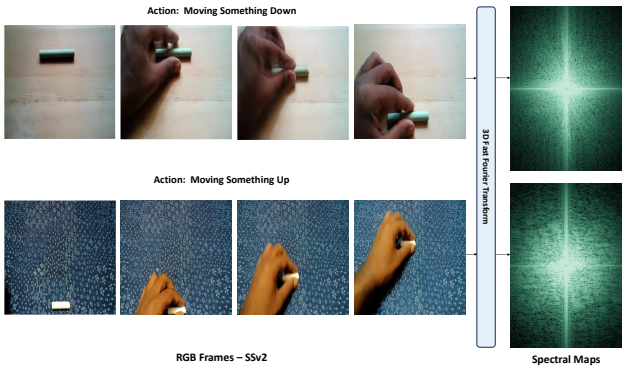

Figure 5. **SSv2 symmetric actions.** Spectral maps (right) reveal clear directional frequency differences between *moving something down* and *moving something up*, despite nearly identical RGB frames.

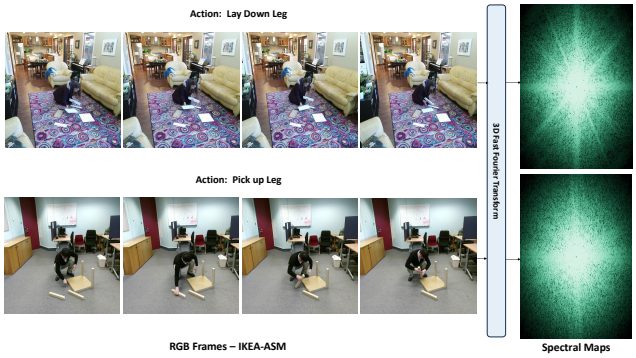

Figure 6. **IKEA-ASM symmetric actions.** 3D FFT spectra (right) distinguish *lay down leg* from *pick up leg* through subtle motion-direction cues that are hard to see in RGB space.

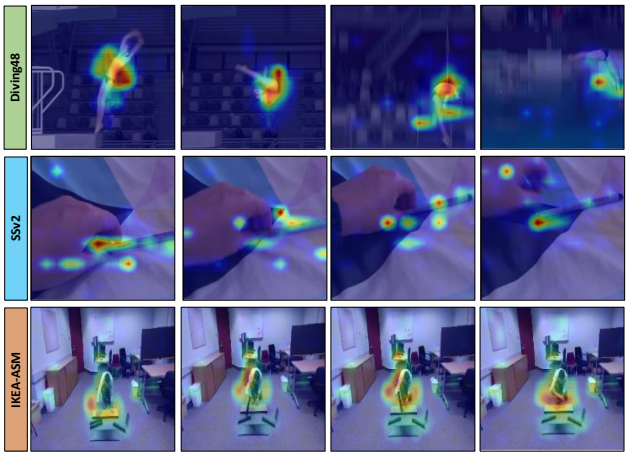

Figure 7. Cross-dataset attention visualizations. Frame-wise attention maps for Diving48, SSv2, and IKEA-ASM show that Frame2Freq consistently localizes motion-critical regions effectively.

The examples from SSv2 (Figure 5) highlight why fine-grained, direction-sensitive actions are so challenging in RGB space. Moving something up and moving something down contain nearly identical hand–object configurations and frame-to-frame appearance. Yet their 3D FFT spectra differ sharply: the frequency maps show clear orientation-specific energy shifts that mirror the direction of motion. These patterns are invisible in raw frames but emerge cleanly in the spectral domain, revealing why frequency cues can reliably disambiguate actions that defeat standard spatial reasoning.

The IKEA-ASM examples (Figure 6 show the same phenomenon in a more complex human–object environment. The actions lay down leg and pick up leg share almost the same spatial layouts and pose transitions, making them nearly symmetric in RGB frames. Their spectral signa-

tures, however, diverge: periodicity and directionality encoded in the 3D FFT expose motion structure that the eye struggles to perceive. Together, the two figures illustrate why frequency-domain representations are uniquely suited for distinguishing fine-grained, nearly mirrored actions.

B.5. Qualitative Analysis

Figure 7 illustrates how Frame2Freq adapts its focus across three very different fine-grained activity datasets. In Diving48, the model concentrates on the diver's torso and limb trajectories, capturing rotation phases that define somersault classes. In SSv2, attention locks onto the fingertips and the manipulated object, revealing the subtle contact dynamics that distinguish nearly identical hand–object motions. In IKEA-ASM, the maps consistently highlight the worker's hands, tools, and assembly components, the regions that drive temporal progression in furniture-assembly actions. Across all settings, the heatmaps show that Frame2Freq reliably isolates the motion-critical regions, demonstrating robust cross-domain temporal grounding.