Fig. S1: LoRA-, DoRA- and SHiRA-based DreamBooth on SDXL. Prompts for style/subject personalization - *left pair*: "A picture of a dog in `<STYLE:WOODEN-SCULPTURE>` style in a bucket" and *right pair*: "A picture of a sunset in `<STYLE:CANVAS>` style". Here, "`<SUBJECT>`" and "`<STYLE>`" are special identifier tokens for style/subject

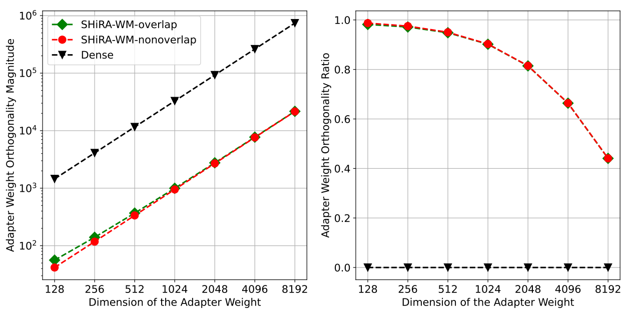

Fig. S2: Adapter Weight Orthogonality Magnitude (AWOM: L2 magnitude) and Adapter Weight Orthogonality Ratio (AWOR: Sparsity Ratio) of the product $\mathcal{A}_1^T\mathcal{A}_2$ between two adapters for unstructured SHiRA-WM overlap and non-overlapping cases (99% sparse). We vary the adapter dimensions (e.g., 4096 refers to a pretrained weight of dimensions 4096×4096) and measure AWOM and AWOR for each weight size (averaged over 50 seeds). For unstructured SHiRA masks, overlapping and non-overlapping adapters achieve *coinciding* AWOR and AWOM, thus suggesting that their orthogonality properties are very similar due to high sparsity. This explains our multi-adapter LLM results.

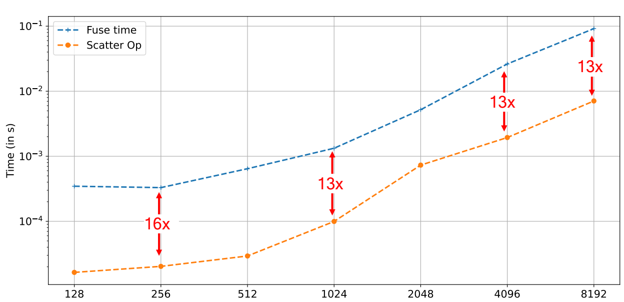

Fig. S3: Reproducing Fig. 7 from Appendix B (submitted paper) in semilogy scale to show clear speedups for all weight dimensions. Comparison between average times for LoRA-fuse and SHiRA-`scatter_op` implmentations on a CPU.

Adapter	**cifar10**	**cifar100**	**food101**	**dtd**
LoRA	97.94	87.97	84.27	69.41
SHiRA	**98.05**	**88.15**	**84.43**	**69.73**

Table S1: LoRA vs SHiRA for Image Classification using ViT-Base model. SHiRA consistently outperforms LoRA on these transfer learning tasks.

Adapter	**#Params**	**COLA**	**QNLI**	**MPRC**	**SST2**	**Average**
LoRA	1.33M	69.73	93.76	89.71	95.57	87.19 (+0%)
SoRA	910K	**71.48**	**94.28**	91.98	95.64	**88.34** (+1.15%)
SHiRA	**636K**	70.62	93.90	**92.15**	**96.50**	**88.29** (+1.10%)

Table S2: GLUE benchmarking for the DeBERTa-V3-base. As evident, with nearly 2x smaller adapter, SHiRA outperforms LoRA by 1.1% accuracy on average. Further, SHiRA achieves a similar accuracy as SoRA while being 30% smaller in adapter size. Hence, SHiRA generalizes to other language tasks as well.

Adapter	**Peak GPU memory (GB)**	**#Training steps/s**
LoRA-PEFT	35.10	**0.69**
DoRA-PEFT	49.49 (+40.99%)	0.49 (-28.98%)
SHiRA-PEFT	**29.26** (-16.63%)	**0.67** (-2.89%)

Table S3: Peak GPU memory consumption (in GBs) and #Training steps per second during training for PEFT-based implementation of various adapters for LLaMA2-7B. Training setup is similar to that used for experiments in section 5.3.1 of the submitted paper. Batch size is 16 for all models and training is done on a single NVIDIA A100 GPU. Relative changes compared to LoRA are highlighted: Green indicates improved performance (lower memory consumption, faster training speed), while Red indicates degraded performance (higher memory consumption, slower training speed). SHiRA trains at nearly the same speed as LoRA but consumes up to 16% lower peak GPU memory.

Model	**LoRA**	**SHiRA**	**Speed-up**
SDXL	3.64 ± 0.10	$\mathbf{0.77 \pm 0.09}$	**4.68×**
LLaMA2-7B	28.15 ± 1.62	$\mathbf{4.93 \pm 0.23}$	**5.71×**

Table S4: End-to-End switching time on CPU for SDXL and LLaMA2-7B: We achieve a very high (4.7×-5.7×) speed up in switching time compared to LoRA.

	Base	**Arc_e**	**BoolQ**	**PIQA**
Base	0	**37.0**	**67.0**	**75.0**
Arc_e		0	75.0	81.5
BoolQ			0	98.5
PIQA				0

Table S5: L2 distances between pretrained base weights and SHiRA adapters vs. distances between adapters: Adapters are closer to the base model weights than to each other.