Appendix

A. Theoretical Analysis: Spectral Bounds

In this section, we provide the formal proof for Theorem 1 stated in the main text and expand upon the connection between spectral norms and the propagation of activation outliers in deep networks.

A.1. Proof of Activation Magnitude Bound

We first restate the bound regarding the relationship between the spectral norm of a weight matrix and the magnitude of the output activations.

Theorem 1 (Restated)**.** *Let $\mathbf{y} = \mathbf{W}\mathbf{x}$ be the output of a linear layer for an input vector $\mathbf{x} \in \mathbb{R}^n$ and weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$. The Euclidean norm of the output vector is bounded by the spectral norm of the weight matrix $\sigma_{\max}(\mathbf{W})$, such that:*

$$\|\mathbf{y}\|_2 \leq \sigma_{\max}(\mathbf{W}) \cdot \|\mathbf{x}\|_2 \tag{7}$$

Proof. Let $\|\cdot\|_2$ denote the Euclidean norm on vectors. The matrix norm induced by the vector Euclidean norm (the spectral norm) is defined as:

$$\|\mathbf{W}\|_2 := \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{W}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sigma_{\max}(\mathbf{W}) \tag{8}$$

where $\sigma_{\max}(\mathbf{W})$ is the largest singular value of $\mathbf{W}$. By the definition of the supremum, for any specific $\mathbf{x} \in \mathbb{R}^n$, it must hold that:

$$\frac{\|\mathbf{W}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \sigma_{\max}(\mathbf{W}) \tag{9}$$

Multiplying both sides by $\|\mathbf{x}\|_2$ (assuming $\mathbf{x} \neq \mathbf{0}$; the trivial case holds for $\mathbf{x} = \mathbf{0}$) yields:

$$\|\mathbf{y}\|_2 = \|\mathbf{W}\mathbf{x}\|_2 \leq \sigma_{\max}(\mathbf{W})\|\mathbf{x}\|_2 \tag{10}$$

□

This result establishes that a large spectral norm is a necessary condition for a linear layer to amplify a reasonably scaled input into a large-magnitude output outlier.

B. Algorithm

Algorithm 1 presents the complete training procedure for Selective Spectral Decay (S^{2}D). The algorithm operates by periodically computing singular value decompositions and selectively penalizing dominant spectral components responsible for activation outliers.

Key steps:

1. **Periodic SVD updates (Lines 6–17):** Every k training steps, the algorithm computes the SVD of each layer's weight matrix $\mathbf{W}^{(l)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, identifying the spectral structure of the weights.
2. **Outlier detection (Line 9):** Using the Principal Component Dominance Ratio (PCDR), the algorithm identifies the minimum rank k where PCDR $\geq \tau$. This determines how many dominant singular values are responsible for creating outliers.
3. **Penalty matrix construction (Lines 11–12):** For layers with identified outliers, a penalty matrix $\mathbf{G}_{\text{reg}}$ is constructed by raising the top-k singular values to power n and reconstructing the partial matrix. This targets only the problematic spectral components.
4. **Gradient update (Lines 18–21):** During each training step, the standard task gradient is augmented with the cached penalty $\lambda\mathbf{G}_{\text{reg}}$, applying selective regularization pressure to the weight components aligned with the largest singular values while leaving other components largely unaffected.

C. Analysis of Outlier Origins

In this section, we expand upon the connection between adaptive optimizers and the formation of activation outliers in transformer models. While the dominant singular *directions* (U, V) encode semantically meaningful representations, their *extreme magnitudes* (Σ) are predominantly optimization artifacts rather than functionally necessary features.

Evidence from Prior Work. Several independent lines of evidence support this characterization. [2] show that Adam-trained models exhibit rapid growth in excess kurtosis (> 100), indicating the emergence of significant outlier channels, whereas SGD-trained models maintain substantially lower kurtosis throughout training. [6] demonstrate that Adam's component-wise normalization privileges the training basis; when this basis is rotated to decorrelate the model, outliers disappear without performance loss, confirming they are not functionally necessary. Furthermore, [7] link outlier features to large diagonal adaptive learning rates in Adam, showing that reducing adaptivity minimizes outlier formation.

Implications for S^{2}D. These findings establish that outlier magnitudes are preventable artifacts of AdamW's basis preference and anisotropic update dynamics. S^{2}D acts as a targeted counter-force to this spectral amplification. The fact that S^{2}D maintains or improves full-precision accuracy (e.g., +1.2% on LLaVA, Table 5 in the main text) confirms that suppressing these extreme magnitudes is benign to the model's semantic capacity.

Algorithm 1 Selective Spectral Decay (S^2D)

Input: Weights $\mathbf{W}$
Hyperparams: Power n, Reg. strength λ, Learning rate η, Update frequency k, PCDR threshold τ
Initialize step counter $t \leftarrow 0$
Initialize penalty matrices $\mathbf{G}_{\text{reg}}^{(l)} \leftarrow \mathbf{0}$ for all layers l
while training do
  if $t \bmod k = 0$ **then** ▷ Periodic spectral update
    for layer $l = 1$ **to** L **do**
      $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow \text{SVD}(\mathbf{W}^{(l)})$
      $\hat{k} \leftarrow \min\{k' : \text{PCDR}(\mathbf{\Sigma}, k') \geq \tau\}$ ▷ Identify outlier rank cutoff
      if $\hat{k}$ is defined **then**
        $\mathbf{\Sigma}_n \leftarrow \text{diag}(\sigma_1^n, \ldots, \sigma_{\hat{k}}^n)$
        $\mathbf{G}_{\text{reg}}^{(l)} \leftarrow \mathbf{U}_{:,1:\hat{k}}\, \mathbf{\Sigma}_n\, (\mathbf{V}_{:,1:\hat{k}})^\top$ ▷ Cache penalty matrix
      else
        $\mathbf{G}_{\text{reg}}^{(l)} \leftarrow \mathbf{0}$ ▷ No significant outlier rank found
      end if
    end for
  end if
  $\nabla_{\mathbf{W}} \mathcal{L}_{\text{task}} \leftarrow \text{Backward}(\mathcal{L}_{\text{task}}(\text{batch}))$ ▷ Standard task loss gradient
  for layer $l = 1$ **to** L **do** ▷ Apply regularized update
    $\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \left(\nabla_{\mathbf{W}^{(l)}} \mathcal{L}_{\text{task}} + \lambda\, \mathbf{G}_{\text{reg}}^{(l)} \right)$
  end for
  $t \leftarrow t + 1$
end while

D. Comparison with Alternative Regularization Approaches

S^2D vs. Rotation Methods (SpinQuant, QuIP). Rotation-based methods mitigate outliers by *redistributing* activation magnitudes through a learned or analytically computed basis transformation $W' = RW$. In contrast, S^2D *suppresses* the spectral cause directly by penalizing the dominant singular values in Σ. These two strategies are thus orthogonal and potentially complementary. Additionally, S^2D avoids the online inference overhead of rotation methods, producing standard weights compatible with vanilla deployment kernels.

S^2D vs. Standard Spectral Regularization. Standard spectral regularization applies uniform pressure to every singular component across the network. As shown in Table 6, applying spectral regularization without the PCDR diagnostic collapses W4A4 accuracy to 40.1% on ImageNet (SigLIP2-Base-384). PCDR acts as a surgical guide, targeting only the spectral components identified to cause pathological activation concentration. Without this selectivity, regularization indiscriminately suppresses both harmful and beneficial spectral components, degrading model capacity.

Table 6. **Impact of PCDR-guided layer selection.** Comparison of S^2D with and without PCDR targeting on ImageNet classification using ERQ quantization (SigLIP2-Base-384). Without PCDR, uniform spectral regularization severely degrades low-bit performance.

	No PCDR	S^2D
W4A4	40.1	**73.0**
W5A5	77.1	**81.9**
W6A6	81.2	**82.7**

E. Self-Supervised Backbone

We extend our experiments to DINOv3 [22], a recently trained self-supervised vision backbone. We attempted to quantize DINOv3 using the official ERQ codebase [35]; however, all ERQ configurations yielded near-random accuracy regardless of bit-width, suggesting an incompatibility with the self-supervised feature distribution. We therefore report results exclusively with PTQ4ViT [33] in Table 8. We also present spectral statistics for DINOv3 in Table 7. Both the PTQ results and spectral patterns are consistent with the SigLIP2 findings, confirming that S^2D improves quantization of modern self-supervised models.

Table 7. **Comparison of FFN activation and weight statistics for SigLIP2 and DINOv3.** We report the $PCDR_1$ and maximum absolute activation of the FFN layers, along with the maximum singular value of their corresponding weights, after fine-tuning with AdamW and AdamW+S^2D.

		$PCDR_1$		Max Abs. Activation		Max Singular Value	
Model	**Layer**	**AdamW**	**S^2D**	**AdamW**	**S^2D**	**AdamW**	**S^2D**
SigLIP2	Layer 5	0.91	0.46	176.40	59.68	10.38	3.22
	Layer 9	0.77	0.09	1166.2	614.7	7.87	3.96
DINOv3	Layer 2	0.47	0.11	1048.41	440.38	37.29	17.94
	Layer 4	0.45	0.22	115.55	67.56	8.67	3.79

Table 8. **PTQ4ViT quantization results on DINOv3-Base** with and without S^2D regularization. ImageNet top-1 accuracy (%) is reported.

Method	**W8A8**	**W6A6**	**W5A5**
AdamW	58.03	**57.04**	28.52
AdamW + S^2D	**76.53**	56.10	**30.69**

F. Extension to Language Models

To evaluate the generality of S^2D beyond vision and vision-language tasks, we conduct a preliminary experiment on a pure language model. We fine-tune Qwen2.5-0.5B using supervised fine-tuning (SFT) on the Dolci dataset, applying S^2D with the same hyperparameters used in the vision experiments (no task-specific tuning). We then evaluate on GSM8K (0-shot) under round-to-nearest (RTN) quantization at various bit-widths. Results are presented in Table 9.

Despite using fewer than 1B training tokens and no hyperparameter adaptation for the language domain, S^2D consistently improves quantized performance across W8A8, W7A7, and W6A6 settings, with gains of +2.2, +2.6, and +2.0 percentage points respectively. The slight reduction in full-precision performance (-1.0) reflects the regularization trade-off, which is more than compensated by the quantization gains. We expect that a learning rate sweep and longer training schedule would further improve both full-precision and quantized results.

Table 9. **GSM8K 0-shot results with RTN quantization for Qwen2.5-0.5B.** S^2D improves quantized accuracy across all tested bit-widths using the same hyperparameters from the vision experiments.

	S^2D	**Baseline**
FP	28.2	29.2
W8A8	**26.2**	24.0
W7A7	**24.7**	22.1
W6A6	**19.8**	17.8

Table 10. **Hyperparameter sensitivity analysis for S^2D on ImageNet.** We vary the SVD computation frequency (k), number of targeted singular values (topk), and PCDR threshold. The **default** configuration is shown in bold.

k	topk	τ	FP16 (%)	W4A4 (%)
Vary SVD update frequency k				
10	3	0.95	85.0	72.2
100	**3**	**0.95**	**85.0**	**73.0**
Vary number of targeted singular values				
100	5	0.95	85.0	72.5
100	10	0.95	84.9	72.6
Vary PCDR threshold τ				
100	3	0.90	84.9	72.4
100	3	0.80	84.8	72.8

G. Hyperparameter Sensitivity

We analyze the robustness of S^2D by varying its key hyperparameters. As shown in Table 10, the default configuration (k=100, topk=3, threshold=0.95) yields the highest W4A4 performance at 73.0%.

We observe that a larger update interval (k=100) outperforms frequent updates (k=10), suggesting that accumulating statistics over a longer horizon improves stability. Interestingly, increasing topk from 3 to 10 results in a marginal performance drop, indicating that outlier mitigation is most effective when targeting only the few most dominant singular directions. Finally, a stricter PCDR threshold of 0.95 proves optimal compared to lower values.

G.1. Sensitivity to Power Exponent n

The power exponent n in S^2D controls the degree of non-uniformity in the penalty applied to singular values: larger n concentrates regularization pressure more aggressively on

Table 11. **Sensitivity to power exponent n.** ImageNet accuracy (%) using ERQ quantization on SigLIP2-Base-384. n=2 provides the best balance across bit-widths.

	Baseline	$\boldsymbol{n=2}$	n=3	n=4
W4A4	65.6	**73.0**	68.3	69.6
W5A5	78.5	**81.9**	79.0	78.7
W6A6	81.1	**82.7**	81.7	82.4

the dominant singular values. We chose n=2 (yielding a σ^3 penalty) to exert stronger regularization on the singular values contributing to outliers while preserving smaller components. Table 11 confirms that while higher orders (n=3, 4) still outperform the baseline, n=2 offers the optimal trade-off between outlier suppression and capacity preservation.

G.2. Amortized SVD Stability

A potential concern with the amortized SVD computation (every m=100 steps) is whether the cached singular vectors (U, V) become stale and lead to inaccurate gradient updates. To validate this design choice, we analyzed the stability of the S^2D gradient penalty computed using cached versus freshly computed SVD factors. The cosine similarity between the two gradient signals remains above 0.99 over the m=100 step caching interval, confirming that the singular vector subspaces evolve slowly relative to the caching frequency. This justifies the computational amortization and explains why k=100 outperforms more frequent updates (k=10) in Table 10: the additional noise from frequent recomputation slightly destabilizes training without meaningful accuracy benefit.