Resting Neurons, Active Insights

Robustifying Activation Sparsity in LLMs via Spontaneity

Haotian Xu[1] · Jiannan Yang[1] · Tian Gao[2] · Tsui-Wei Weng[3] · Tengfei Ma[1]

[1]Stony Brook University [2]IBM T.J. Watson Research Center [3]UC San Diego

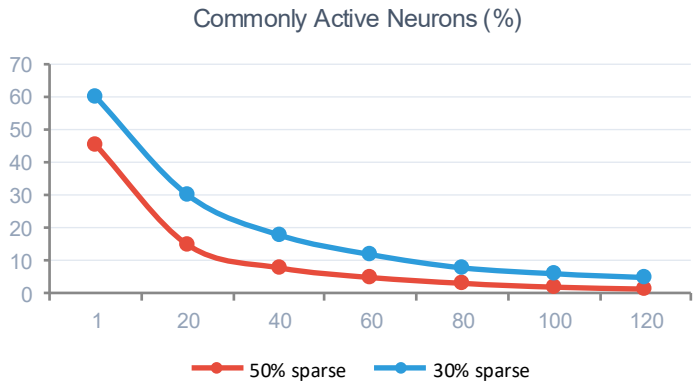
ICML 2026

The Problem: Activation Sparsity Breaks Representations

The Challenge

- Activation sparsity reduces LLM inference cost by zeroing low-magnitude activations
- At high sparsity (≥50%), severe accuracy degradation occurs across all benchmarks
- Root cause: representational instability — sparsity disrupts input-dependent patterns learned during pretraining
- Neurons commonly active across all tokens decay exponentially with sequence length

Key Observation



Insight: The absence of stable, universally active neurons leads to distributional shifts in hidden states

Biological Inspiration: Spontaneous Neural Activity

Biological Systems

- Neurons exhibit baseline firing even without external stimuli
- Spontaneous activity encodes prior expectations
- Stabilizes downstream responses and provides persistent reference
- Well-documented since Hubel & Wiesel (1962)

Our Approach: SPON

- Inject learnable, input-independent activation vectors
- Act as persistent representational anchors
- Complement dynamically selected neurons under sparsity
- Distill global, input-agnostic info from dense model

How SPON Works

1

Sparsify Inputs

Apply threshold masking to input activations:
$Y = W \cdot S(X)$

2

Add Spontaneous Activation

Inject learnable vector α:
$Y = W \cdot S(X) + W \cdot \alpha$

3

Train via KL Divergence

Align sparse output distribution with dense model

4

Absorb into Bias

Fold $W \cdot \alpha$ into bias:
Zero inference overhead

Core equation: $Y = W \cdot S(X) + W \cdot \alpha$ where α is input-independent and absorbed into bias after training

Theoretical Foundation: Fisher-Weighted Correction

Optimality Condition

Define sparsity residual: e(X) = WX − WS(X)

Minimizing KL divergence yields:

$$\mathbb{E}[\, W^T H (W \cdot \alpha - e(X)) \,] = 0$$

where H is the Hessian (= Fisher Information Matrix)

SPON approximates the sparsity residual under a Fisher-weighted metric, prioritizing directions where output probabilities are most sensitive.

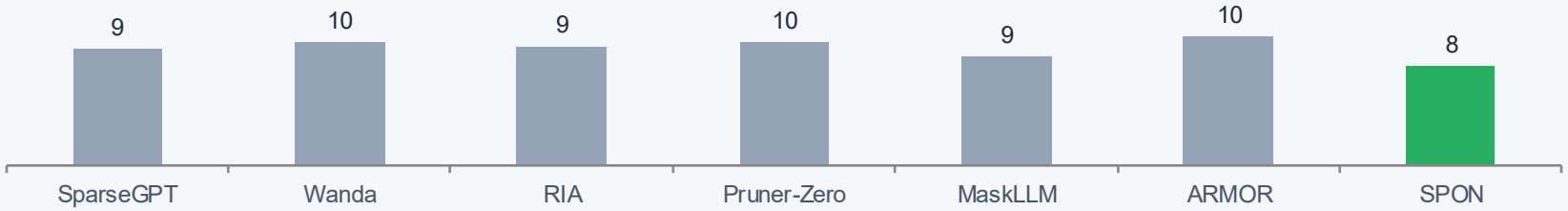
Key Implications

- Corrections lie in the pretrained feature subspace of W
- Fisher weighting focuses on high-sensitivity directions
- Minimal complexity overhead (only d params per module)
- No overfitting risk due to massive N vs. tiny param count

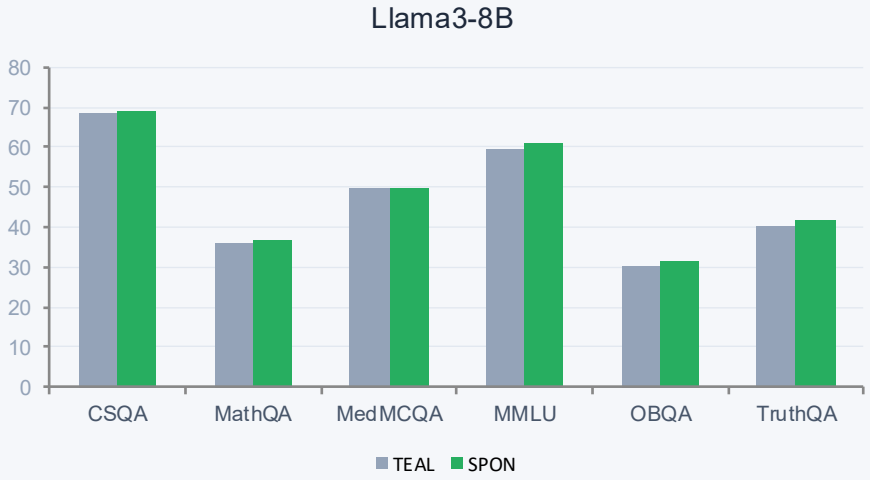
Results: Language Modeling (Perplexity ↓)

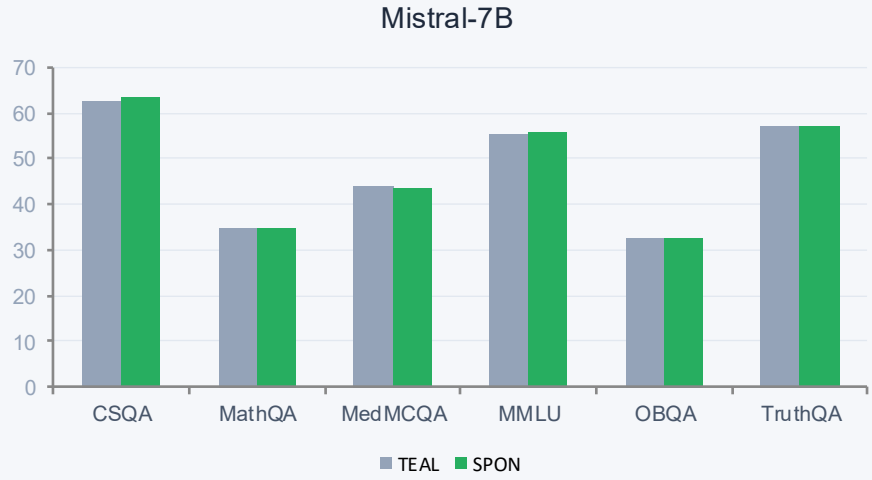
Method	Sparsity	Llama3-1B	Llama3-8B	Mistral-7B	Qwen3-8B
Full	0%	12.14	6.75	5.49	8.99
TEAL	50%	17.03	8.34	6.00	9.75
SPON	50%	**15.43**	**7.83**	**5.86**	**9.26**
TEAL	60%	—	11.62	6.90	11.38
SPON	60%	—	**9.63**	**6.51**	**10.42**

SPON vs. Network Pruning at 50% Sparsity (Llama3-8B)



Results: Zero-Shot Generalization at 50% Sparsity





1.178%

Avg. relative improvement

0.016%

Additional parameters

0

Extra inference FLOPs

Comparison with SOTA Methods at 50% Sparsity

Backbone	Method	ARC-C	ARC-E	BoolQ	PiQA	Wino	Avg
Llama3-8B	LaRoSA	48.04	74.75	77.55	79.98	68.98	69.82
	WINA	48.81	75.00	81.25	79.16	70.64	70.97
	R-Sparse	46.42	76.94	76.73	79.92	67.80	69.56
	WAS	47.06	78.83	77.49	78.63	70.52	70.51
	SPON	**51.11**	**76.89**	**82.54**	78.40	**70.88**	**71.96**

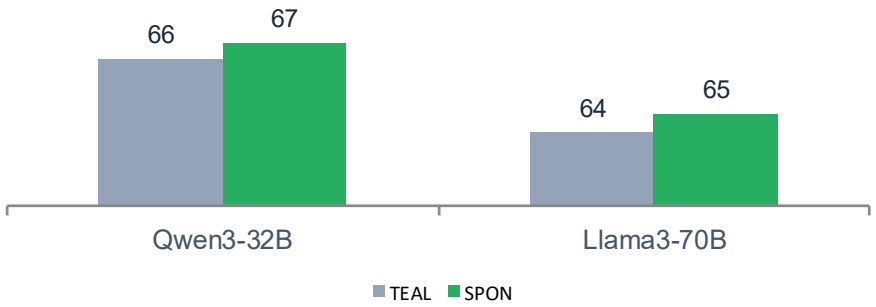
SPON is Orthogonal to Existing Methods

SPON addresses the representational drift from activation masking — a distinct problem from rotation-based (LaRoSA), weight-informed (WINA), rank-aware (R-Sparse), or Bayesian (WAS) approaches.

These methods could potentially be combined with SPON for further gains.

Scalability & Inference Efficiency

Scale-Up: Large Models



Throughput (tok/s)

Method	L3-8B	M-7B
Full (0%)	42.65	44.17
TEAL (50%)	64.28	71.25
SPON (50%)	**64.13**	**71.84**

Broad Compatibility

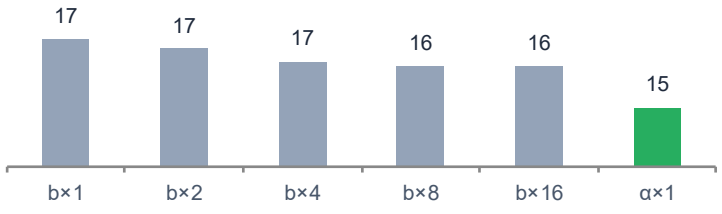
MoE Models
Improves OLMoE-7B & DeepSeek-16B under reduced expert budgets

Quantization
Stable across 4/8/16/32-bit; int4 + SPON beats full 32-bit baseline

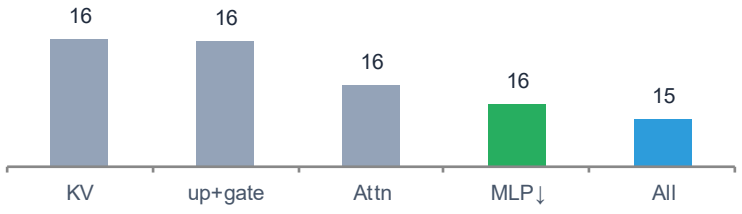
Weight Pruning
SPON + Wanda pruning yields consistent gains across backbones

Ablation Studies: Design Choices Matter

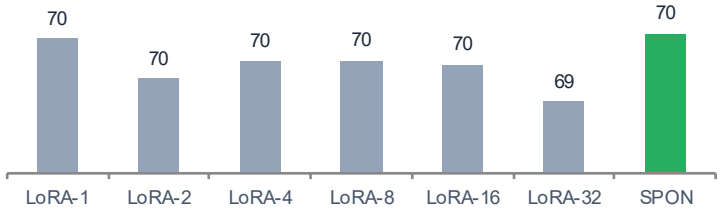
Learning from α vs. bias b



Where to Inject SPON



SPON vs. LoRA (Accuracy %)



Key Takeaways

- A single α neuron outperforms 16 bias vectors — activation-space learning is essential
- MLP down projections are the sweet spot — matching full injection with 0.006% params
- Lower layers benefit most, consistent with localized semantic encoding
- SPON beats LoRA at 50× fewer parameters

Understanding Why SPON Works

CKA Representational Alignment

Centered Kernel Alignment (CKA) measures how well sparse model representations match the dense model layer by layer.

SPON consistently achieves higher CKA similarity to the dense model than TEAL alone across all 32 decoder layers.

Diagonal entries reach >95% similarity with SPON vs. losing resemblance with TEAL.

Off-diagonal patterns also better match the dense model's self-CKA structure.

Latent Space Drift (t-SNE)

Llama3-8B

TEAL L_2: **8.69** SPON L_2: **8.21**

Mistral-7B

TEAL L_2: **13.33** SPON L_2: **12.82**

Qwen3-8B

TEAL L_2: **6.76** SPON L_2: **6.24**

SPON substantially reduces the distributional shift introduced by activation sparsity, providing a mechanistic explanation for improved knowledge retention.

Conclusion & Takeaways

Principled Solution

SPON stabilizes activation-sparse LLMs by mitigating representational drift through Fisher-weighted correction

Minimal Overhead

Only 0.016% extra parameters, zero additional FLOPs — absorbed into bias terms at inference

Universal Gains

1.178% avg improvement across architectures (1B–70B), benchmarks, and sparsity levels

Broad Compatibility

Orthogonal to existing methods; synergizes with MoE, quantization, and weight pruning

Deeper Insight

Challenges the assumption that bias-like components are redundant — they serve as crucial representational anchors under sparsity

Code: github.com/hxu105/SPON