ZeroTuning: Unlocking the Initial Token's Power to Enhance Large Language Models Without Training

Feijiang Han[1], Xiaodong Yu[1,2], Jianheng Tang[3], Delip Rao[1], Weihua Du[4], Lyle Ungar[1]

[1]University of Pennsylvania, [2]AMD, [3]Peking University, [4]Carnegie Mellon University

TL;DR

- **The Problem:** Existing training-free attention tuning methods are complex and biased, relying on heuristics to find "important" task-specific tokens.
- **Our Insight:** Don't search for complex solutions. The most powerful control lever is universal and already there: the **initial token (e.g., <BOS>).**
- **Our Solution (ZeroTuning):** A simple, few-line code modification to precisely tune the initial token's attention, requiring zero parameter updates, and working in supervised and unsupervised modes.
- **The Impact:** ZeroTuning achieves significant gains across **15 datasets**, outperforming previous, more complex methods.

METHOD

The methodology consists of two key steps:

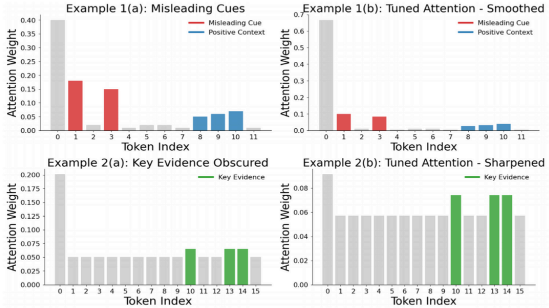
- **Head Behavior Profiling:** Categorizing heads into up-effective (performance improves with more initial token's attention) and down-effective
- **Selective Rescaling:** Conducting supervised or unsupervised searches to get scaling factors for attention scores or key states

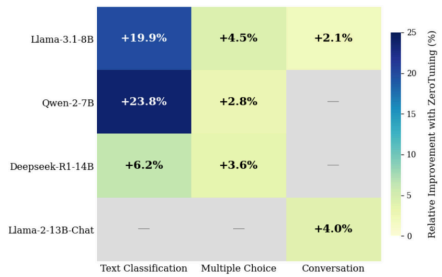
```
Class LlamaAttention(nn.Module):
  def forward(self, target_layers, target_heads, scaling_factor, ...):
      # ... omitting unmodified LlamaAttention code
      # 1. Standard attention weight calculation
      attn_weights = F.softmax(torch.matmul(query_states,
                          key_states.transpose(2, 3)), dim=-1)
      # 2. Our [ZeroTuning] Method
      if self.layer_idx in target_layers:
          # Shape: (bsz, num_heads, q_len, kv_len)
          attn_weights[:, target_heads, :, 0] *= scaling_factor
          # Re-normalize the Attention
          attn_weights[:, target_heads] =
          F.normalize(attn_weights[:, target_heads], p=1, dim=-1)
      # 3. Compute attention output
      attn_output = torch.matmul(attn_weights, value_states)
      # omitting unmodified LlamaAttention code ...
```

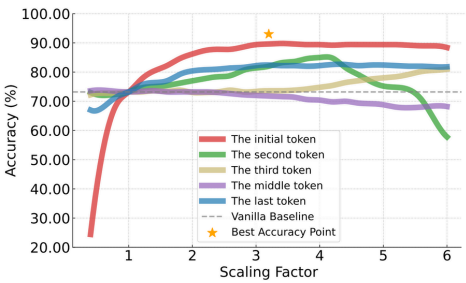
INSIGHTS

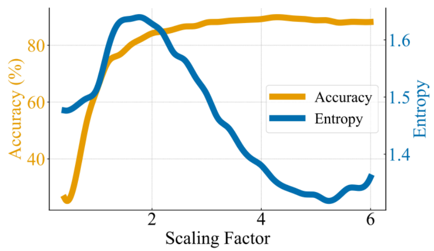
The Initial Token Tuning Effect

- It sharpens or smooths attention focus, an effect amplified by the token's natural "attention sink" role.
- This boosts accuracy by correcting pretrained biases and reduces uncertainty (output entropy) for more confident predictions.
- Crucially, minimum entropy aligns with maximum accuracy, enabling a powerful unsupervised tuning method.









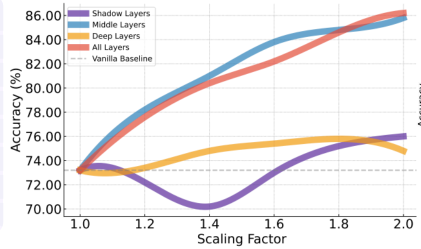
DEEPER ANALYSIS

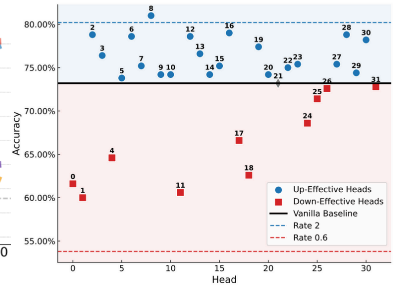
Tuning Which Layers:

- Greatest impact from shallow & middle layers.
- Optimal performance by jointly tuning all layers..

Tuning Which Heads:

- Attention heads show distinct tuning preferences.
- Selectively tuning the dominant head type （up or down effective） outperforms uniform tuning.





Dataset	Method	Extra Context Length 0	100	200	300	Average
SST-2	Vanilla	73.20	68.40	59.20	32.00	58.20
	ZeroTuning	**91.60**	**89.20**	**87.40**	**85.40**	**88.40**
	Diff	18.40	20.80	28.20	53.40	30.20
BoolQ	Vanilla	69.60	68.60	67.60	68.60	68.60
	ZeroTuning	**82.40**	**81.80**	**81.40**	**81.20**	**81.70**
	Diff	12.80	13.20	13.80	12.60	13.10
LogiQA	Vanilla	39.40	36.60	36.20	35.80	37.00
	ZeroTuning	**42.40**	**43.00**	**41.00**	**41.00**	**41.85**
	Diff	3.00	6.40	4.80	5.20	4.85
PIQA	Vanilla	83.60	82.20	81.20	80.60	81.90
	ZeroTuning	**85.40**	**83.80**	**83.20**	**82.80**	**83.80**
	Diff	1.80	1.60	2.00	2.20	1.90

Shot	Method	SST-5	BoolQ	MMLU	AQUA	Average
0-Shot	Vanilla	45.4	69.6	67.4	25.7	52.0
	ZeroTuning	**52.0**	**82.4**	**68.80**	**30.4**	**58.40**
	Diff	6.6	12.8	1.4	4.7	6.4
1-Shot	Vanilla	47.6	80.4	61.8	28.1	54.5
	ZeroTuning	**49.4**	**82.4**	**63.4**	**30.0**	**56.3**
	Diff	1.8	2.0	1.6	1.9	1.8
2-Shot	Vanilla	50.4	83.4	64.4	25.7	56.0
	ZeroTuning	**52.4**	**85.0**	**66.0**	**32.8**	**59.1**
	Diff	2.0	1.6	1.6	7.1	3.1

Works With:

- Longer Contexts
- In-context Learning (Few-shot)
- Resource Constraints
- Diverse Decoding Strategies
- Prompt Variations
- Quantized Models (4/8-bit)