Let Attention Control Its Sharpness: A Per-Head Query-Gain Scalar

Vuk Rosic

June 14, 2026

Abstract

We add a single learnable scalar per attention head that controls how sharp or diffuse that head's attention distribution is. The scalar multiplies the query vector by $(1 + g)$ before the attention score is computed, so it acts as a direct gain on the softmax logits. Initialized at zero, it begins training as the exact baseline and lets each head choose its own sharpness during training. On a 7.7M-parameter language model trained on 20M tokens, this adds only 144 parameters and no new matrix multiplications, yet improves validation loss by -0.078 (from 4.7984 to 4.7200) at fixed seed and configuration.

1 Introduction

Standard scaled dot-product attention fixes the temperature of the softmax for every head: each head must distribute its attention with the same baseline sharpness, set implicitly by the $1/\sqrt{d_{\text{head}}}$ scaling. Yet different heads plausibly want different behaviour — some should commit to a single token, others should average broadly over context.

We ask a minimal question: *if we give each head one scalar to set its own sharpness, does it help?* The change is deliberately tiny so that any improvement is attributable to the mechanism alone, not to added capacity.

2 Method

Let $Q, K \in \mathbb{R}^{T \times d_{\text{head}}}$ be the per-head query and key matrices. The baseline score is

$$\text{score} = \frac{QK^\top}{\sqrt{d_{\text{head}}}}.$$

We introduce one learnable scalar g per head and scale the query before the score is formed:

$$Q' = Q\,(1 + g), \qquad \text{score}' = \frac{Q'K^\top}{\sqrt{d_{\text{head}}}} = \text{score}\,(1 + g).$$

Thus g is a direct multiplicative gain on the softmax logits. The $(1 + g)$ parameterization means $g = 0$ leaves the model exactly equal to the baseline, so training starts from an unchanged forward pass and learns to turn the dial only where it helps:

$g < 0 \;\rightarrow$ flatter (averages many tokens), $\quad g = 0 \;\rightarrow$ baseline, $\quad g > 0 \;\rightarrow$ peaked (commits to one token).

Implementation. The mechanism is one zero-initialized parameter vector of length n_{heads} per layer:

```
self.q_gain = nn.Parameter(torch.zeros(self.n_heads))  # one per head
Q_head = Q_head * (1.0 + g_head)
```

On the model used here (6 heads $\times$ 24 layers) this is 144 scalars and zero additional matrix multiplications. We are not adding capacity; we are giving an existing degree of freedom a place to turn.

3 Experimental Setup

All runs use the `screen10m` configuration: a $\sim$ 7.7M-parameter decoder LLM trained on 20M tokens, fixed seed 42, identical optimizer, schedule, and data ordering. The only difference between conditions is the presence of the per-head query gain. We report final validation loss.

4 Results

The query-gain model trains stably and ends below the baseline throughout the run. Figure 1 shows the validation curves; Figure 2 shows the final values.

Condition	Final val loss	Δ
Control (baseline)	4.7984	—
Per-head Q-gain (+144 params)	4.7200	-0.0784

Table 1: Final validation loss at 20M tokens, seed 42.

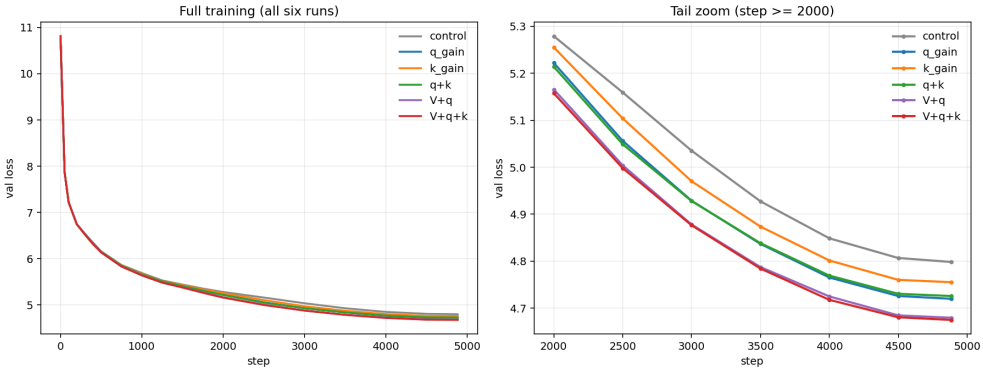

Figure 1: Validation loss over training. The per-head query gain stays below the baseline for the full run.

5 Discussion

The effect is small in absolute terms but large per parameter: -0.078 validation loss for 144 scalars and no extra compute. The baseline pins every head to a single sharpness; the gain lets each

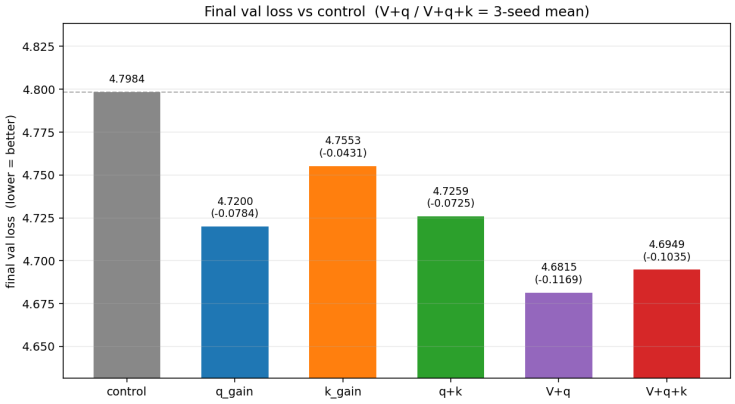

Figure 2: Final validation loss. A -0.078 improvement for 144 added scalars.

head pick its own, and the learned values spread across heads rather than collapsing to zero. The mechanism is orthogonal to architecture changes and can be applied to queries, keys, or both.

6 Conclusion

A single learnable sharpness scalar per head — zero-initialized so the model begins as the exact baseline — is a cheap, safe knob that improves a small LLM's validation loss at negligible cost. The lesson is not the magnitude of one number but the principle: give attention an explicit control over its own sharpness, per head, and let training decide.