

Multi-Head mHC: No Gain at Equal Parameters

Marcos Hernanz

marcos@hernanz.io

Abstract

Manifold-constrained hyper-connections (mHC) widen the residual stream into several copies that each layer reads from, writes to and mixes, with coefficients shared by all channels. Multi-head mHC gives each group of channels its own read, write and doubly stochastic mix while keeping the guarantees of mHC. In 629 training runs at 27M and 112M parameters, we tune the learning rate of every main variant and compare against parameter-matched controls. Multi-head mHC gives no gain at equal parameters. With the parameter count of mHC it is no better than mHC, and when it adds parameters it is no better than a wider feed-forward layer. At 27M this holds with QK-norm and, at one learning rate, with four times the tokens. At the best learning rate of mHC it also holds for each tested variant that makes only some of the read, write and mix per group. At a shared learning rate tuned for mHC in a pilot, heads that add parameters beat mHC, but a wider feed-forward layer with the same parameters gains as much. Without QK-norm the heads degrade more gently at too high a learning rate, where at 27M every variant fails through attention-logit growth. QK-norm removes this growth and with it the advantage of the heads. Apart from a smoother gradient norm that does not show in the loss, we find no evidence that multi-head mHC would make larger models more stable.

Contents

1	Introduction	3
2	Related Work	5
3	Method	7
3.1	Background on mHC	7
3.2	Multi-head mHC	7
3.3	Guarantees	8
3.4	A depth view	8
3.5	Predictors and parameter cost	8
3.6	Ablations and other variants	9
4	Experimental Setup	10
4.1	Models, data and training	10
4.2	Baselines, controls and learning rates	10
4.3	Statistics and hardware	11
5	Results	12
5.1	Comparison at equal parameters	12
5.2	Longer training	14
5.3	Variants that add parameters	14
5.4	Behavior above the optimal learning rate	15
5.5	Effect of QK-norm	17
5.6	Ablation of the per-head blocks	19
6	Discussion and Limitations	21
6.1	Comparing connection variants	21
6.2	Stability at larger scale	21
6.3	Why the heads do not help	22
6.4	Limitations	23
7	Conclusion	24
	References	25
A	Comparability Across Chips	27
B	Learning-Rate Grids	28
C	Implementation and Compute	29

1 Introduction

Hyper-connections (HC) widen the residual stream of a transformer into n parallel copies and let every layer learn how to read from them, write to them and mix them [49]. Manifold-constrained hyper-connections (mHC) make this stable at scale by projecting the mixing matrix onto the doubly stochastic matrices, so that the stream can neither blow up nor lose its mean through depth [40]. In both designs every connection coefficient is a scalar that multiplies a whole D -dimensional vector, so all channels of the stream take the same route through depth.

Multi-head attention showed that several smaller attention heads, each with its own projections, can work better than a single head of full width at a similar cost [36]. The same split applied to mHC gives multi-head mHC (§3.2). The D channels are cut into h groups, and each group gets its own read, write and doubly stochastic mix. The design keeps the guarantees of mHC, group by group (§3.3), and seen through depth it gives each group of channels its own pattern of connections to earlier layers (§3.4).

This paper asks whether multi-head mHC gives a better model for the same parameters and compute. We ran a controlled study at 27M and 112M parameters, 629 training runs on TPU v5e and v6e. Runs are paired by seed, every model that adds parameters is compared with what the same parameters buy in a wider feed-forward layer, and every main variant has a learning-rate grid at both widths. At 27M we also ablate the per-head blocks (read, write and mix), train with

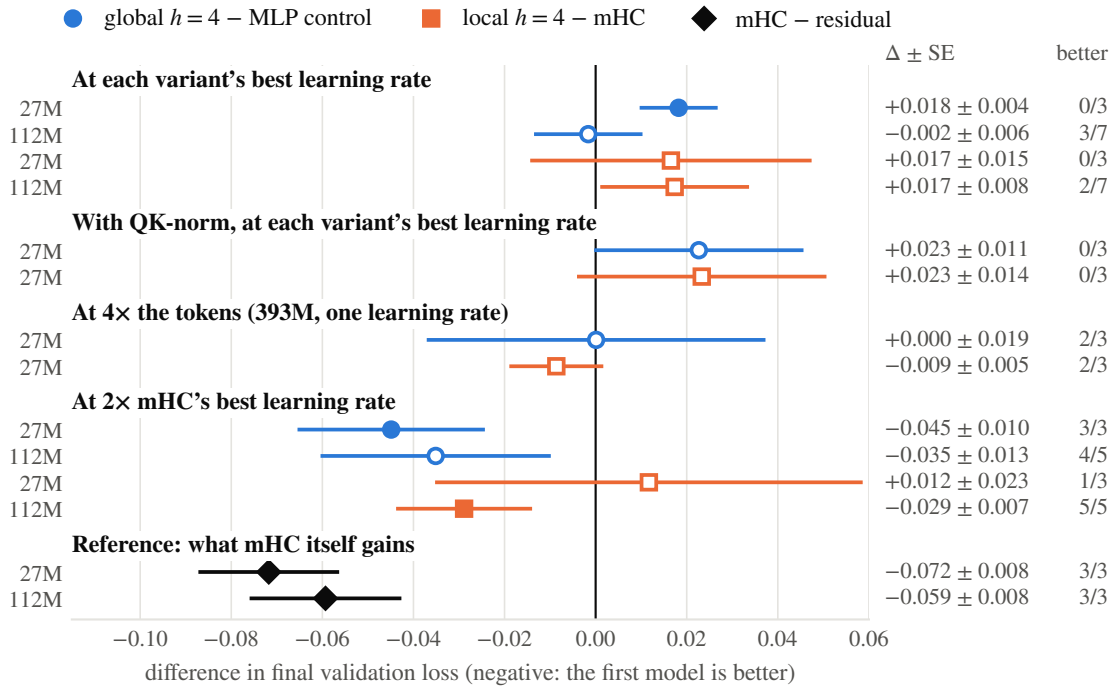


Figure 1: Summary of the main comparisons. Local heads have the parameter count of mHC. Global heads add parameters and are compared with the MLP control, mHC with its feed-forward layer widened to the same count (§3.5 and §4.2). Each row is a difference in final validation loss with the standard error of §4.3, shown as the mean ± 2 SE over seeds. Filled markers show established effects, with $|\text{mean}| > 2$ SE and all seeds agreeing in sign. The columns on the right give the mean $\pm$ SE and the number of seeds on which the first model is better. The second group of rows uses QK-norm and the others do not. The 112M rows other than mHC – residual pool two TPU generations (Appendix A). With every model at its own best learning rate the heads gain nothing at equal parameters. They lead only above the optimal learning rate, and only without QK-norm (§5.4 and §5.5).

four times the tokens and repeat the grid with QK-norm, which normalizes the queries and keys of each attention head [13]. Decision rules were written down before the results they decide. They are public, with the code and the results of every run (Appendix C).

The answer is no (Figure 1). With every model at its own best learning rate, heads at the parameter count of mHC are no better than mHC, and heads that add parameters are no better than mHC with a feed-forward layer widened to the same count. This holds at both widths, and at 27M it holds with QK-norm and, at one learning rate, with four times the tokens. At 27M and the best learning rate of mHC, no subset of the three per-head blocks that we tested uses its parameters better than the feed-forward layer (§5.6). Compared in the more common way, at one shared learning rate and without a parameter-matched control, the heads that add parameters beat mHC by about 40% of mHC’s gain over the residual connection, at $h = 4$ on 9 of 10 seeds. It takes both controls to see that this gain comes from the extra parameters and the choice of learning rate (§5.3 and §6.1).

The heads change how training fails when the learning rate is too high. Without QK-norm they degrade less than mHC just above the optimal learning rate, and robustness of this kind, measured in smaller models, has been used as a proxy for the instabilities of large runs [30, 38]. We tested whether this tolerance amounts to a stability benefit and found no evidence that it does (§5.4, §5.5 and §6.2). At 27M every variant fails above the optimum through attention-logit growth, and the heads do not prevent it. QK-norm, which current open models use, removes the failure, and with QK-norm the heads are more sensitive to the learning rate than mHC. In this test only one run, which fails, has loss spikes. The one difference in favor of the heads that survives QK-norm is that their gradient norm crosses the clipping threshold less often than that of mHC. This does not show in the loss, and whether it would matter in a large model is untested.

2 Related Work

Most follow-up work on HC and mHC re-parameterizes the single $n \times n$ mix over the copies [7, 12, 18, 22, 42, 48] or the dense maps that predict the coefficients [11]. Other work raises the number of copies while updating only a few of them per sublayer [46] or carries a static form of mHC to state space models [25]. Some designs use channel groups as the streams themselves [50] or widen the stream and split it into such groups [6], with one shared mix in both. None of them gives each group of channels its own copies and its own mix. In a trained mHC model a typical site uses only about two of its four copies [47], and when the copies start identical they can collapse onto a dominant one [1].

The value of the mix itself has been questioned. Several recent designs drop it [9, 17, 30], and a concurrent submission finds that learned read and write gates with the mix fixed to the identity match mHC [4]. Another shows that constraining one marginal of the mixes already bounds the norm of their product at any depth [3].

Table 1 lists the designs closest to ours by where their coefficients are shared. The gated residual of Qwen3.8-Next [30] reads a four-branch stream through an elementwise gate and writes with one scalar per branch, without a mix. Its read has the granularity of ours at $h = D$. The ablations in that work find the granularity of the read more useful than that of the write, and the mix of little value once the read and write are expressive. SiHC [17] uses static per-channel read and write maps, with the mix fixed to the identity, in pixel diffusion transformers whose streams hold different subpatches. uHC [5], a concurrent submission, argues, as our design assumes, that readout patterns shared across feature dimensions are a bottleneck. It computes the read of each group of channels from that group’s own slice of the copies, which is our per-head read with a local predictor (§3.5), and adds a low-rank write-back with extra parameters while keeping the mix of mHC. Its validation loss is lower than that of mHC by 0.006 and 0.008 at 46M and 128M parameters and by 0.128 at 363M, where its mHC baseline is 0.093 worse than the residual connection. In that work all methods share one learning rate per scale, and no seed count is reported. Section 5.6 tests the per-head read alone, with a global predictor, against the gain that the same parameters bring in the feed-forward layer. The per-group mix is the one part of our design that none of these has.

Table 1: Where the connection coefficients are shared in the designs closest to multi-head mHC. “Shared” means the same coefficients for all channels. Each mix is an $n \times n$ matrix over the copies, doubly stochastic in mHC and in our design.

design	read	write	mix
HC [49], mHC [40]	shared	shared	shared
Qwen3.8-Next gated residual [30]	per channel	shared	none
SiHC [17]	per channel, static	per channel, static	identity
uHC [5], concurrent	per group	shared plus low-rank	shared
multi-head mHC (ours)	per group	per group	per group

The same split has been applied to other connections. Multi-Head Attention Residuals [21] split the depth-attention query of Attention Residuals [14] into channel groups with independent routers. Their loss is U-shaped in the number of heads with its minimum at 4–8. On web data, with the learning rate tuned for each method, the split beats a single query at 100M–1B parameters, by 0.01 at 100M and 0.17 at 1B, and on a second corpus, at a shared learning rate, by at most 0.03. An anonymous submission with the same title [2] adds temperature-matched controls, which attribute most of the gain of the split over a single query to the flatter softmax of each smaller head

and only a small part to routing each subspace separately. The report that introduced Attention Residuals tested the same split with 16 heads and found a slightly higher loss (1.752 against 1.746), concluding that the best mixture over depth is largely uniform across channels [14]. In Depth-Attention [43], per-head softmax mixing over depth improves only slightly on a fixed uniform average of the same layers (2.2115 against 2.2153 at 500M parameters), and Kimi K3 [15] ships single-head Attention Residuals.

Small-scale comparisons of residual variants depend on how they are controlled. A per-channel residual gate compared against parameter-matched baselines was within noise up to 320M parameters, significant at 590M but only a trend at 1B [16]. With the learning rate tuned per architecture at 400M parameters, the best learning rate of mHC equals that of the pre-norm residual [37]. More broadly, most of the transformer modifications that Narang et al. [26] reimplemented in one shared code base did not meaningfully improve performance, and well-regularized standard LSTMs outperformed more recent recurrent architectures in language modeling once all of them were tuned by the same large-scale hyperparameter search [23]. We also tune the learning rate of each main variant and find that this choice decides whether the heads appear to beat mHC (§5.1 and §5.3).

Raising the learning rate reproduces in small models two instabilities first seen in large ones [38]. In the first, attention logits grow until attention collapses onto one key. This appeared at about 8B parameters while vision transformers were scaled to 22B [8] and is known as attention entropy collapse [44]. In the second, the output logits drift. A normalization of the queries and keys, QK-norm, removes the first, and a z-loss, a penalty on the squared log of the softmax normalizer of the output, removes the second. QK-norm was introduced for translation [13], was applied in a LayerNorm form to attention-logit growth by Dehghani et al. [8] and is now standard in open models [10, 35, 41]. Qwen3.8-Next stress-tests its new architecture and optimizer against what they replace at two to four times the optimal learning rate and counts loss spikes and clipped steps [30]. We use the same measures to ask whether the tolerance of the heads to a too-high learning rate is a stability benefit (§5.4, §5.5 and §6.2).

3 Method

3.1 Background on mHC

Per token, the residual stream is $X \in \mathbb{R}^{n \times D}$, made of n copies of width D ($n = 4$ unless stated otherwise). A sublayer F , either attention or feed-forward, is wrapped as

$$u = H^{\text{pre}} X, \quad y = F(u), \quad X' = H^{\text{res} \top} X + H^{\text{post} \top} y, \quad (1)$$

with a read $H^{\text{pre}} \in \mathbb{R}^{1 \times n}$, a write $H^{\text{post}} \in \mathbb{R}^{1 \times n}$ and a mix $H^{\text{res}} \in \mathbb{R}^{n \times n}$. In mHC the three are predicted per token from the RMS-normalized, flattened stream by three linear maps, each scaled by a learned α and added to a static bias. The read and write are $H^{\text{pre}} = \sigma(\cdot)$ and $H^{\text{post}} = 2\sigma(\cdot)$, with σ the logistic sigmoid and $(\cdot)$ the predicted values, and the mix is $H^{\text{res}} = \text{Sinkhorn}(\exp(\cdot))$, which is doubly stochastic up to the accuracy of the Sinkhorn iteration [40]. A doubly stochastic matrix has spectral norm at most 1 and preserves the sum over the copies, and a product of such matrices is again doubly stochastic. The mix composed over any number of layers therefore stays bounded.

3.2 Multi-head mHC

We split the D channels into h contiguous groups of D/h channels, which we call heads, and give each group its own read, write and doubly stochastic mix. Equation (1) is applied to the $n \times D/h$ block of the stream that belongs to group g , $X_{(g)}$, and to the matching block of the sublayer output, $y_{(g)}$,

$$u_{(g)} = H_{(g)}^{\text{pre}} X_{(g)}, \quad y = F([u_{(1)}, \dots, u_{(h)}]), \quad X'_{(g)} = H_{(g)}^{\text{res} \top} X_{(g)} + H_{(g)}^{\text{post} \top} y_{(g)}. \quad (2)$$

The sublayer is unchanged and takes as input the concatenation of the group inputs $u_{(g)}$. With $h = 1$ the model is mHC. Figure 2 shows one sublayer of each.

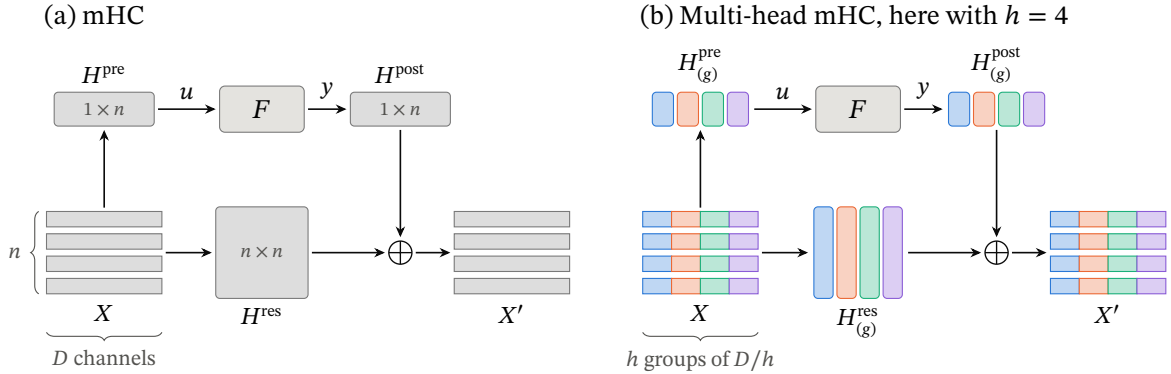


Figure 2: One sublayer of mHC (a) and of multi-head mHC (b). mHC reads the n copies into the sublayer input u , mixes them and writes the output y back, with the same coefficients for all D channels. Multi-head mHC gives each of the h channel groups (shown in different colors) its own read, mix and write. The sublayer F is the same in both.

3.3 Guarantees

On $\text{vec}(X)$, ordered by group, then copy, then channel, the mixing operator of Eq. (2) is block diagonal,

$$\text{diag}(H_{(1)}^{\text{res}\top} \otimes I_{D/h}, \dots, H_{(h)}^{\text{res}\top} \otimes I_{D/h}). \quad (3)$$

Its spectral norm is the largest of those of its blocks, which is at most 1. Each block preserves the sum over copies of its own channels, and a product of such operators is again block diagonal with doubly stochastic blocks. All three properties of mHC therefore hold, group by group.

3.4 A depth view

Unrolling Eq. (1) shows that sublayer l reads a weighted sum of the embedding and all earlier outputs,

$$u_l = \sum_{s < l} c_{ls} y_s, \quad c_{ls} = H_l^{\text{pre}} H_{l-1}^{\text{res}\top} \dots H_{s+1}^{\text{res}\top} H_s^{\text{post}\top}, \quad (4)$$

where y_0 is the embedding. The plain residual has $c \equiv 1$. DenseFormer learns such weights over the outputs of earlier blocks [27], and MUDDFormer predicts them from the hidden state [39]. In mHC the copies act as a state of size n carried through depth, which is written by H^{post} , moved by H^{res} and read by H^{pre} . The matrix c is therefore n -semiseparable, meaning that every block below its diagonal has rank at most n . mHC gives all D channels one such pattern, and h heads give each channel group its own. Raising n instead raises the rank of the single pattern, at the cost of a wider stream.

3.5 Predictors and parameter cost

How the coefficients of each group are produced is a separate choice. A *local* predictor computes the coefficients of group g from its own $n \times D/h$ slice of the RMS-normalized stream and has the parameters of mHC, cut into h blocks. A *global* predictor computes the coefficients of every group from the whole stream and has h times the parameters of the mHC predictor. A *static* predictor uses the biases only. Table 2 gives the parameter counts. At $D = 384$, global $h = 4$ adds 1.8M parameters to mHC (6.5%) and $h = 16$ adds 9M (33%), so every model with global heads needs a control with the same parameter count (§4.2).

Table 2: Parameter counts of the main models at the two widths. An MLP control is mHC with its feed-forward layer widened to the parameter count of the model it controls (§4.2).

model	$D = 384$	$D = 768$
residual connection	26.75M	110.12M
mHC	27.34M	111.90M
local $h = 4$	27.34M	111.90M
global $h = 4$	29.13M	117.25M
MLP control for global $h = 4$ (mHC, SwiGLU width 1216 / 2240)	29.11M	117.21M
global $h = 16$	36.29M	–
MLP control for global $h = 16$ (mHC, SwiGLU width 1992)	36.26M	–

3.6 Ablations and other variants

Any subset of the three blocks (read, write and mix) can be made per head while the others are predicted once and shared by all groups. With no block per head the model is exactly mHC, and with only the read per head and a local predictor it is the group-wise readout of uHC [5]. We tested two side variants. The first is a joint $(nh) \times (nh)$ doubly stochastic mix over (copy, group) pieces, which lets groups exchange channels. The second is mHC with $n = 8$ copies. We did not test a mixture of several Sinkhorn matrices, since the doubly stochastic matrices form a convex set and such a mixture can express nothing that a single mix cannot.

4 Experimental Setup

4.1 Models, data and training

The models are pre-norm GPTs [31] with rotary attention [34], SwiGLU feed-forward layers [33] and an untied output layer (Table 3). They are trained on FineWeb-Edu [28] with a 16,384-token BPE vocabulary [32], using random 1024-token windows from a 200M-token pool, and validated on a fixed set of 819k tokens. Training uses AdamW [19] with $\beta = (0.9, 0.95)$, weight decay 0.1 on matrices, 200 warmup steps, cosine decay to 10% of the peak learning rate and gradient clipping at 1.0. Each run takes 6000 steps of 16×1024 tokens, or 98M tokens. The connection runs in float32 and the rest of the model in bf16. mHC follows the published recipe, with $n = 4$, 20 Sinkhorn iterations, α initialized to 0.01 and, as in HC [49], an initialization close to the pre-norm residual.

For the stability test of §5.4 and §5.5, the five main variants at 27M (§4.2) are also trained with QK-norm, an RMSNorm [45] on the queries and keys of each attention head before the rotary embedding. Gemma 3 and Qwen3 also use QK-norm [10, 41], and OLMo 2 normalizes the query and key projections with RMSNorm [35]. These runs use nine learning rates from 1.5×10^{-3} to 24×10^{-3} spaced by a factor of about $\sqrt{2}$, and the grid without QK-norm is extended to the same top. These runs, the extension and the replays of §5.4 log the loss and the gradient norm before clipping at every step instead of every 25th. Every 250 steps they log the largest attention logit and the mean attention entropy of every layer on four fixed validation sequences.

Table 3: The two model sizes and their training. At 112M on v6e the residual has only 3 seeds at its best learning rate.

	27M	112M
width D / layers / attention heads	384 / 8 / 6	768 / 12 / 12
SwiGLU width	1024	2048
tokens	98M, and 393M for one test	98M
learning rates ($\times 10^{-3}$)	1.5, 2.1, 3, 4.2, 6	0.53, 0.75, 1.06, 1.5, 2.1
stability test ($\times 10^{-3}$)	1.5 to 24, QK-norm on and off	–
seeds per learning rate	3 (v6e), 2–10 (v5e)	3 (v6e, 5 at the best), 2 (v5e)
TPU	v6e and v5e	v6e and v5e

4.2 Baselines, controls and learning rates

Every main comparison includes mHC and the plain residual connection. Global heads add parameters, so global $h = 4$ and $h = 16$ are each compared with mHC widened in the feed-forward layer to the same parameter count, which we call the *MLP control* (Table 2). Local heads have the parameter count of mHC and are compared with mHC directly. For other variants we use the *MLP line*, which is piecewise linear in the parameter count and passes through mHC and its widened versions. It gives the loss that the same parameters buy in the feed-forward layer.

The first experiments used a single peak learning rate, 3×10^{-3} , which was the best for mHC in a pilot. We then mapped the learning rate of the five *main variants*, mHC, global and local $h = 4$, the MLP control and the residual, so that each can be compared at its own best. The best learning rate is the one with the lowest mean final loss over seeds 0–2 (0–1 where only two seeds were run), a rule fixed in advance.

4.3 Statistics and hardware

For a given seed, every variant sees the same batches and, except for the MLP controls, starts from the same weights outside the connection, so differences at the same learning rate are paired by seed. We report the mean difference in final validation loss, first model minus second, with its standard error and the number of seeds on which the first model is better. We call an effect *established* when $|\text{mean}| > 2 \text{ SE}$ and all seeds agree in sign, a rule fixed before the main runs. Comparisons between variants at different learning rates, and all comparisons of variants at their own best learning rates at $D = 768$, use the two-sample standard error. Where both chips ran the same comparison, we pool the two by inverse variance as independent estimates.

Each of the 629 runs uses one TPU chip, either v5e or v6e. The two chips are not interchangeable, since the same run on v6e behaves like a new seed of the v5e run. Each experiment therefore stays on one chip type with its own baselines, and we never compare numbers from different chips directly (Appendices [A](#) and [C](#)).

5 Results

5.1 Comparison at equal parameters

We compare local heads with mHC and global heads with the MLP control (§4.2), with every variant at its own best learning rate, which is the fair setting, and at one learning rate shared by all, which is the usual setting for connection variants.

The learning-rate grids put the optimum of mHC, global $h = 4$, the MLP control and the residual at 2.1×10^{-3} at 27M on TPU v6e. Local $h = 4$ is best at 3×10^{-3} , ahead of its value at 2.1×10^{-3} by 0.003 (Figure 3, left). At 112M the optimum is 0.75×10^{-3} for mHC, local $h = 4$ and the control. Global $h = 4$ and the residual are best at 1.06×10^{-3} , ahead of their values at 0.75×10^{-3} by 0.0008 and 0.0016 (Figure 3, right). The v5e grid, which has the residual only at 1.5×10^{-3} , puts the optimum of every other variant in the same place (Appendix B). The heads therefore move the optimum by at most one grid point, and they change the range of learning rates within 0.02 of each variant’s own best by at most one point. At 27M this range covers two grid points for mHC and both head variants and three for the control and the residual. At 112M it covers three points for global $h = 4$, two for mHC, the control and the residual, and one for local $h = 4$.

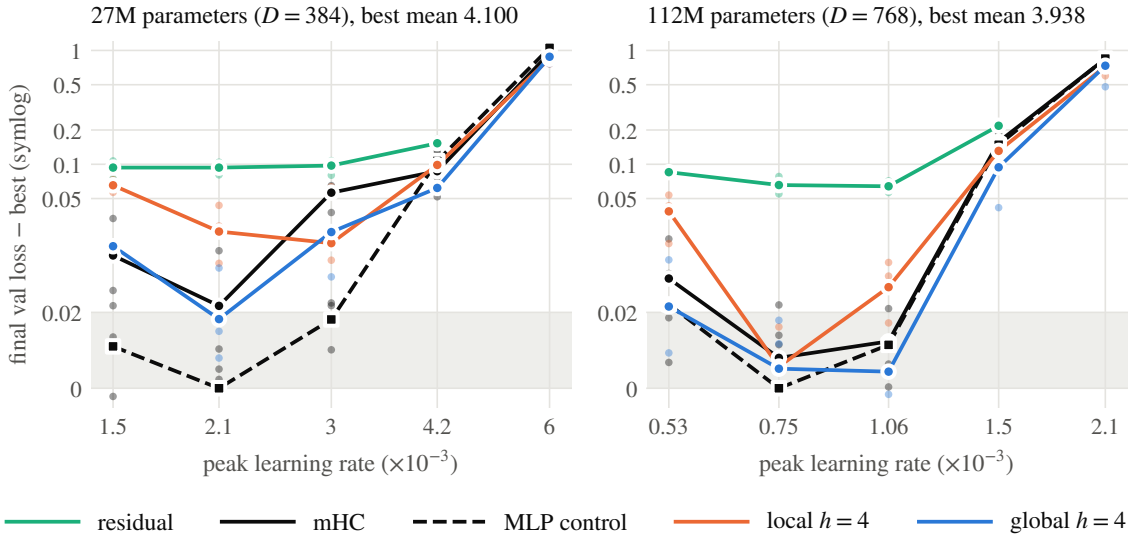


Figure 3: Final validation loss above the best mean at each width, against the peak learning rate (TPU v6e, 98M tokens). Lines show the mean over seeds 0–2 and dots show single seeds. The axis is symlog and linear below 0.05, and the shaded band marks losses within 0.02 of the best variant. The range of good learning rates in the text is measured from each variant’s own best. The heads move the optimum and the range of good learning rates by at most one grid point. At twice the optimum of mHC, 4.2×10^{-3} and 1.5×10^{-3} , global $h = 4$ lies below mHC and the control.

Table 4 compares the variants at these optima. At 27M the MLP control is ahead of global $h = 4$ on every seed, by 0.018 ± 0.004 , so at their best the heads are worth less than their parameters. At 112M the two are level on both chips, and pooled over 7 seeds the difference is -0.002 ± 0.006 . Local heads are no better than mHC at either width ($+0.017 \pm 0.015$ and $+0.017 \pm 0.008$). The gain of mHC over the residual holds at both widths (-0.072 ± 0.008 and -0.059 ± 0.008) and at every learning rate of both grids up to twice the optimum.

The result is the same at a shared learning rate. At 3×10^{-3} , the pilot rate of mHC, on TPU v5e (Table 5), local heads differ from mHC by -0.003 at $h = 2$ (10 seeds) and $+0.003$ at $h = 4$ (6 seeds),

Table 4: Each variant at its own best learning rate. Global and local denote $h = 4$, and control denotes the MLP control. At 27M (TPU v6e, seeds 0–2) differences are paired where the two learning rates coincide. At 112M they use two-sample standard errors, with 5 seeds on v6e (3 for the residual) and 2 on v5e, pooled by inverse variance. * marks established effects.

comparison	27M	112M		
	v6e	v6e	v5e	pooled
global – control	$+0.018 \pm 0.004^*$ (0/3)	$+0.005 \pm 0.007$ (1/5)	-0.016 ± 0.010 (2/2)	-0.002 ± 0.006
local – mHC	$+0.017 \pm 0.015$ (0/3)	$+0.013 \pm 0.011$ (2/5)	$+0.022 \pm 0.012$ (0/2)	$+0.017 \pm 0.008$
global – mHC	-0.003 ± 0.013 (1/3)	-0.005 ± 0.008 (3/5)	-0.007 ± 0.009 (1/2)	-0.006 ± 0.006
control – mHC	-0.022 ± 0.011 (3/3)	-0.010 ± 0.010 (4/5)	$+0.009 \pm 0.013$ (0/2)	-0.004 ± 0.008
mHC – residual	$-0.072 \pm 0.008^*$ (3/3)	$-0.059 \pm 0.008^*$ (3/3)	–	–

and static heads gain nothing over static mHC (-0.003 ± 0.006). Global $h = 4$ is level with its control over 10 seeds ($+0.005 \pm 0.007$), so a gain of the heads over the control larger than about 0.01 lies outside two standard errors. Global $h = 16$ trails its control on every seed ($+0.046 \pm 0.004$), so the same parameters gain 3.4 times as much in the feed-forward layer.

QK-norm does not change the comparison (§5.5). On a nine-point grid at 27M, with every variant at its own best learning rate, local $h = 4$ trails mHC by 0.023 ± 0.014 and global $h = 4$ trails the control by 0.023 ± 0.011 , better on 0 of 3 seeds in each case. A planned repeat at 112M could not be run (§6.4).

Table 5: All connection variants at one shared learning rate. Final validation loss relative to mHC at $D = 384$ and 3×10^{-3} (TPU v5e, 98M tokens), paired by seed and given as mean $\pm$ SE. The last column counts the seeds on which the variant beats the reference. * marks established effects.

model	params	seeds	Δ loss	better
residual connection	26.75M	4	$+0.045 \pm 0.006^*$	0/4
static mHC	26.75M	3	$+0.023 \pm 0.010^*$	0/3
static $h = 4$	26.75M	3	$+0.020 \pm 0.008^*$	0/3
static $h = 8$	26.75M	3	$+0.024 \pm 0.014$	0/3
joint mix, $h = 4$, static	26.75M	3	$+0.013 \pm 0.008$	1/3
local $h = 2$	27.34M	10	-0.003 ± 0.009	5/10
local $h = 4$	27.34M	6	$+0.003 \pm 0.008$	3/6
local $h = 8$	27.34M	3	$+0.010 \pm 0.017$	1/3
local $h = 16$	27.35M	3	$+0.003 \pm 0.010$	1/3
global $h = 2$	27.94M	4	-0.020 ± 0.009	3/4
global $h = 4$	29.13M	10	-0.019 ± 0.006	9/10
global $h = 8$	31.52M	4	-0.021 ± 0.011	3/4
global $h = 16$	36.29M	6	-0.019 ± 0.005	5/6
joint mix, $h = 4$, global	33.85M	3	-0.019 ± 0.014	2/3
mHC, $n = 8$	30.70M	3	-0.006 ± 0.011	2/3
MLP control for global $h = 4$	29.11M	10	-0.024 ± 0.009	8/10
MLP control for global $h = 16$	36.26M	6	$-0.065 \pm 0.007^*$	6/6
global $h = 4$ – its control		10	$+0.005 \pm 0.007$	2/10
global $h = 16$ – its control		6	$+0.046 \pm 0.004^*$	0/6

5.2 Longer training

All results so far use 98M tokens, or 3.6 tokens per parameter at $D = 384$. To test the effect of training length, we trained mHC, global and local $h = 4$, the MLP control and the residual on 393M tokens (14 per parameter) at 3×10^{-3} , with seeds 0–2 on TPU v5e. These runs keep the same warmup and stretch the cosine decay over the longer run, and the comparisons were fixed beforehand. Each token of the pool is seen about twice, and every variant sees the same batches. The 98M-token runs of the same variants, chip, learning rate and seeds provide the shorter length (Table 6).

Table 6: Comparisons at four times the tokens. $D = 384$ at 98M and 393M tokens (TPU v5e, 3×10^{-3} , seeds 0–2), paired by seed. The last column gives the per-seed change from 98M to 393M tokens, which is negative when the first model gains relative to the second with longer training. * marks established effects.

comparison	98M tokens	393M tokens	change
global $h = 4$ – MLP control	$+0.017 \pm 0.024$ (1/3)	$+0.000 \pm 0.019$ (2/3)	-0.017 ± 0.011 (2/3)
local $h = 4$ – mHC	-0.006 ± 0.008 (2/3)	-0.009 ± 0.005 (2/3)	-0.003 ± 0.005 (2/3)
global $h = 4$ – mHC	$-0.024 \pm 0.005^*$ (3/3)	$-0.023 \pm 0.007^*$ (3/3)	$+0.001 \pm 0.011$ (1/3)
MLP control – mHC	-0.041 ± 0.024 (2/3)	-0.023 ± 0.023 (2/3)	$+0.018 \pm 0.001^*$ (0/3)
mHC – residual	$-0.041 \pm 0.007^*$ (3/3)	$-0.048 \pm 0.007^*$ (3/3)	$-0.007 \pm 0.003^*$ (3/3)

Neither of the two comparisons of interest changes by an established amount. At 393M tokens global $h = 4$ and its control are tied ($+0.000 \pm 0.019$), and local heads are within noise of mHC (-0.009 ± 0.005). Two changes are established. The gain of mHC over the residual grows from 0.041 to 0.048, and the lead of the control over mHC shrinks by 0.018 ± 0.001 on every seed while the heads keep theirs. Relative to the control, the heads gain 0.017 ± 0.011 with the longer training, on 2 of 3 seeds, which is not established. On these seeds the lead of the control over the heads at 98M tokens (0.017) is larger than its ten-seed value (0.005), so part of the shift is a return to the mean, and a single longer point with three seeds cannot show a trend. If anything, the shared learning rate favors the heads over mHC. On v6e at 98M tokens it lies above the optimum of every variant except local $h = 4$, a regime in which the tolerance of the heads helps them (§5.4).

5.3 Variants that add parameters

Compared with mHC alone, global heads look like an improvement. At 3×10^{-3} on TPU v5e they beat mHC by about 0.02 at every h from 2 to 16 (Table 5), roughly 40% of the gain of mHC over the residual (0.045). At $h = 4$ the lead holds on 9 of 10 seeds. The global predictor adds 1.8M parameters at $h = 4$, however, and the MLP control gains as much with the same 1.8M in the feed-forward layer (-0.024 ± 0.009 against mHC). Figure 4 places every variant that adds parameters against the MLP line. None lies below it by more than twice its standard error, and every way of spending parameters on the connection beyond $h = 4$, whether more heads, the joint mix or $n = 8$ copies, lies above it once it costs more than a few percent. In the TPU layout of Appendix C, every head count runs at the speed of mHC on v5e (28.4k–29.3k against 28.7k tokens per second per chip in the sweep over h), as do the MLP controls (29.4k and 29.7k against 28.6k), so this is also a comparison at equal compute. With each model at its own best learning rate, even the raw lead over mHC disappears (-0.003 ± 0.013 at 27M and -0.006 ± 0.006 at 112M, Table 4). The gain at 3×10^{-3} therefore needs both the extra parameters, without which local heads gain nothing there, and a learning rate above the optimum, which 3×10^{-3} is for mHC and global $h = 4$ in the v6e grid (§5.4).

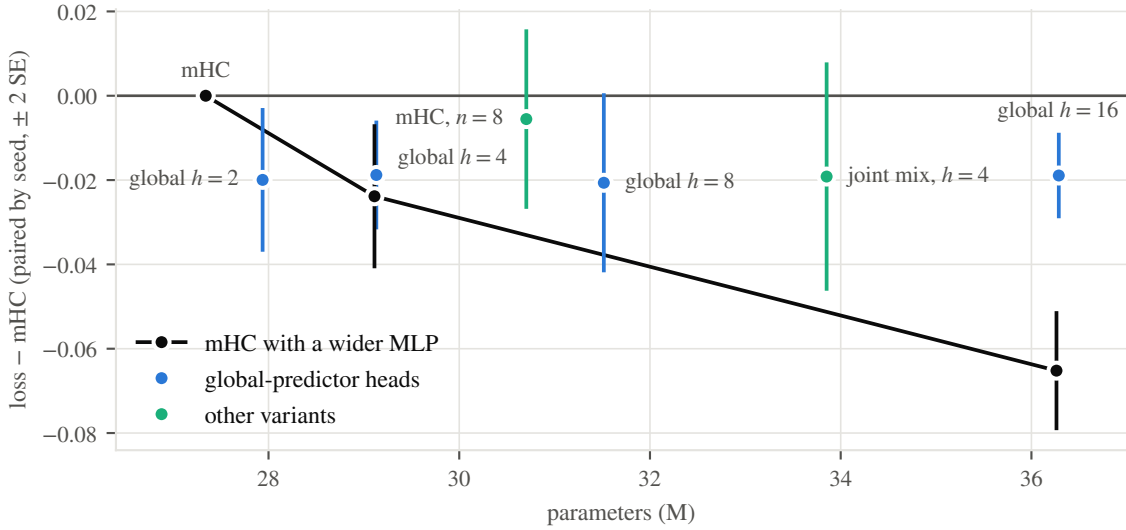


Figure 4: Loss relative to mHC of every variant that adds parameters, against its parameter count ($D = 384$, 3×10^{-3} , TPU v5e, paired by seed, ± 2 SE). The black line is the MLP line, given by mHC with a wider feed-forward layer, and points above it use their parameters less well than the feed-forward layer does. No point lies below the line by more than twice its standard error. Global heads gain about 0.02 over mHC at every h , whether they add 0.6M parameters or 9M, while the feed-forward layer turns the same 9M into a gain of 0.065. Variants that add less than 0.1M parameters are listed in Table 5.

5.4 Behavior above the optimal learning rate

The heads move the optimum and the range of good learning rates by at most one grid point (§5.1), but at the next grid point above all of these ranges, where every variant is at least 0.04 worse than its own best, they degrade less than mHC and the control. At 27M and 4.2×10^{-3} , twice the optimum, global $h = 4$ is 0.045 ± 0.010 ahead of its control (3 of 3 seeds, Figure 1), although it trails the control by 0.018–0.026 at every lower learning rate of the grid (Figure 3, left). At 112M and 1.5×10^{-3} , again twice the optimum, global and local $h = 4$ lead mHC by 0.063 and 0.026 on every seed on v6e, and by 0.068 and 0.069 on v5e. This is the only grid point between these ranges and failure. At the next learning rate, 6×10^{-3} at 27M and 2.1×10^{-3} at 112M, every variant run there ends 0.7–1.1 above its own best, and from 8.5×10^{-3} at 27M runs diverge or end more than 1 above the best. The heads therefore do not make a higher learning rate usable. They fail at the same learning rate as mHC and degrade less just before it. One of our early 112M results was an instance of this tolerance. At 1.5×10^{-3} on TPU v5e, half the 27M rate and untuned, global $h = 4$ led its control by 0.034 ± 0.013 (2 of 2 seeds), but that rate is twice the optimum and costs mHC, the heads and the control 0.09–0.18. Robustness to the learning rate in small models has been proposed as a proxy for the instabilities of large runs [38]. We therefore asked what fails, whether there are loss spikes to count, and whether the tolerance survives QK-norm.

To see what fails, we replayed the 27M runs at 2.1×10^{-3} , 4.2×10^{-3} and 6×10^{-3} with per-step logs and diagnostics (§4.1). All 42 replays reproduce the losses of the original runs bit for bit. Above the optimum, attention logits grow until the attention heads of most layers put almost all of their weight on one key. This is the first of the two instabilities that Wortsman et al. [38] reproduce in small models [see also 8, 44]. At step 1000 the largest attention logit is 47–77 at 2.1×10^{-3} , 470–720 at 4.2×10^{-3} and 1930–2710 at 6×10^{-3} (Figure 5, middle left), and the lowest mean attention entropy of any layer falls from 1.1–1.8 nats to 0.04–0.07 and then to 0.01. This happens in every variant, and the heads slow it only slightly. At 6×10^{-3} , global and local $h = 4$ reach 1930 and

2000, against 2070 for mHC, 2710 for the control and 2470 for the residual. One layer, which we had not planned to examine, behaves differently. As in the residual, the first attention layer of both head variants stays almost uniform at these three learning rates (entropy 4.8–5.9 nats at 6×10^{-3} , averaged over seeds), while that of mHC and the control collapses with the rest (0.24 nats).

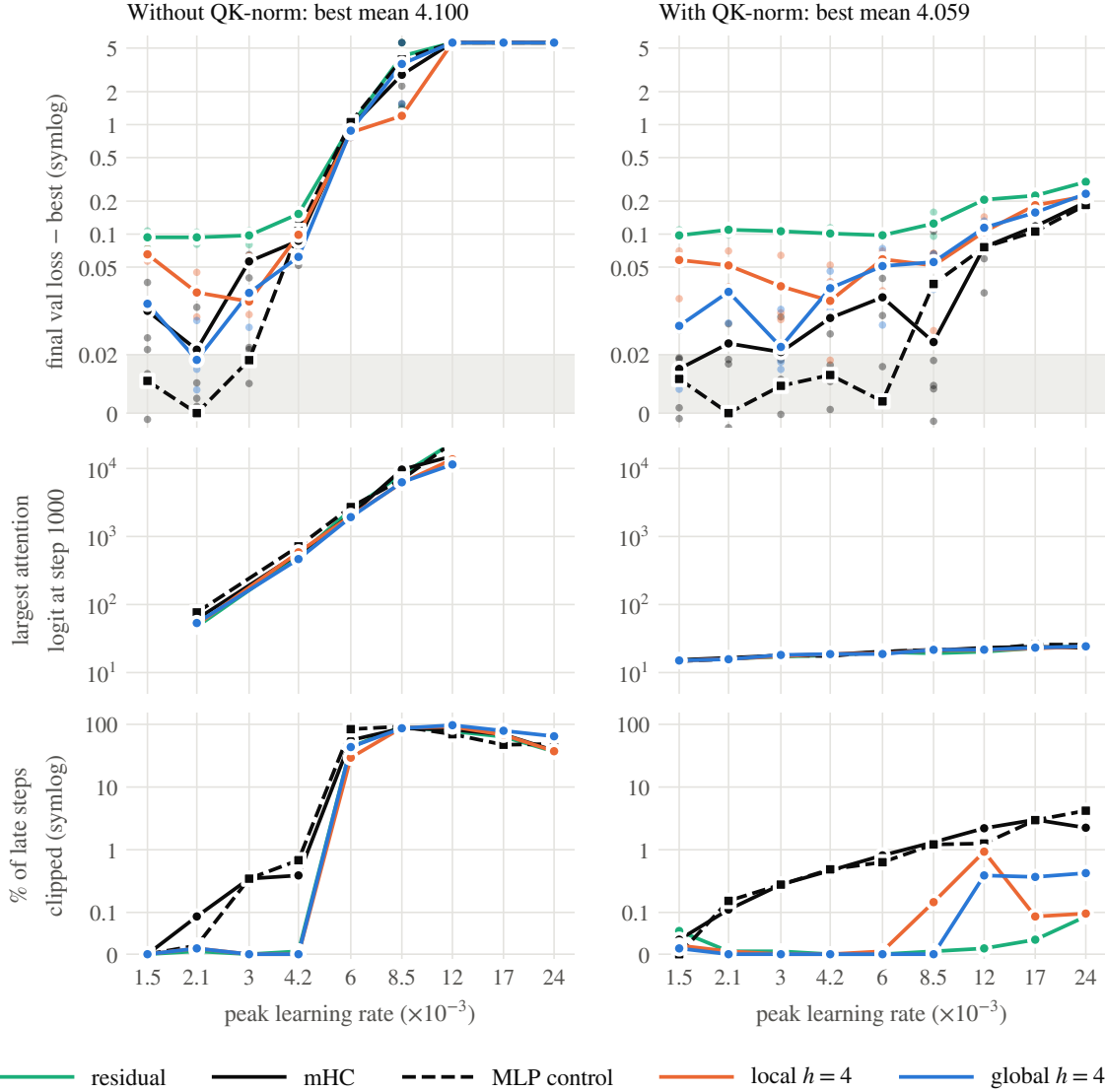


Figure 5: The stability test at 27M, without QK-norm (left) and with it (right), at $D = 384$ on TPU v6e. The top row shows the final validation loss above the best mean of each panel against the peak learning rate. Lines show the mean over seeds 0–2, or over the seeds that ran where runs are missing, and dots show single seeds. The axis is symlog and linear below 0.05, and a diverged run counts as $\ln 16384$. The middle row shows the largest attention logit over layers at step 1000, averaged over the seeds that reach that step. It nearly coincides for all variants. Without QK-norm it lacks 1.5×10^{-3} and 3×10^{-3} , which were not replayed with diagnostics, and the rates above 12×10^{-3} , where every run diverges before step 1000. The bottom row shows the share of steps in the last 80% of training whose gradient norm before clipping exceeds 1.0, averaged over seeds, from every 25th step for the runs at 1.5×10^{-3} and 3×10^{-3} without QK-norm. QK-norm removes the failure and the tolerance of the heads, but not their smoother gradient norm.

We count loss spikes from the per-step logs in the ways used by OLMo 2 and Qwen3.8-Next [30, 35]. OLMo 2’s spike score is the share of steps at least seven standard deviations away from the mean of the previous 1000. By this measure only one run has loss spikes, seed 1 of mHC without QK-norm

at 8.5×10^{-3} , on 0.1% of its steps, and that run ends 1.4 above the best loss of mHC. No other run with per-step logs that lasts beyond step 1000 has any, with or without QK-norm. The same score for the gradient norm is below 0.5% wherever training does not fail, and it is similar for all variants. The count used by Qwen3.8-Next, steps more than 0.1 above a rolling median over 201 steps, does not separate the variants. At a batch of 16k tokens, step-to-step noise crosses that threshold on about 8% of steps at every learning rate where training completes, including the optimum.

We also asked whether the tolerance without QK-norm reflects slower connection dynamics. On TPU v6e, scaling the learning rate of every connection parameter of mHC by 0.25 changed nothing at 3×10^{-3} , 4.2×10^{-3} or 6×10^{-3} . Scaling that of global $h = 4$ by 4 strengthened the tolerance instead of removing it, and global $h = 4$ then ended 0.20 better than before at 6×10^{-3} (± 0.05 , 3 of 3 seeds). Because the decoupled weight decay of AdamW scales with the learning rate, this setting also shrinks the weights of the predictor four times faster and pushes the connection toward a static one. In a follow-up that we designed after this result, and which is therefore exploratory, mHC with the same setting gained nothing at 6×10^{-3} ($+0.030 \pm 0.041$). At the optimum the setting changed neither model by more than the noise, and global $h = 4$ stayed behind its control ($+0.007 \pm 0.005$, 0 of 3 seeds). The tolerance therefore belongs to the head split, and a faster connection strengthens it only when the split is present.

5.5 Effect of QK-norm

With QK-norm no run diverges on the grid, which reaches 24×10^{-3} , sixteen times its lowest learning rate. The largest attention logit at step 1000, averaged over seeds, stays between 15 and 26, and at 24×10^{-3} every variant ends 0.18–0.21 above its own best (Figure 5, right). QK-norm also lowers the best loss of every variant, by 0.037–0.048. Above the optimum the heads now degrade more than mHC. We measure this with the learning-rate sensitivity of Wortsman et al. [38], the mean over the grid of the final loss above the model’s own best, with each loss capped at the loss at initialization, taken as $\ln 16384$. With QK-norm, global and local $h = 4$ are more sensitive than mHC by 0.015 ± 0.002 and 0.006 ± 0.003 , on each of the 3 seeds (Table 7).

Without QK-norm the same measure is dominated by 8.5×10^{-3} , where runs diverge on some seeds and not on others, and each divergence adds about 0.47 to the sensitivity of a seed. Over the five learning rates up to 6×10^{-3} , where no run diverges, global and local $h = 4$ are less sensitive than mHC without QK-norm, by 0.018 ± 0.013 (2 of 3 seeds) and 0.028 ± 0.010 (3 of 3), but not with it ($+0.004 \pm 0.001$ and $+0.000 \pm 0.005$). The rule we fixed before these runs compares the sensitivity over all nine learning rates. It counts the tolerance as removed if, with QK-norm, neither head variant is established as less sensitive than mHC and each difference has the opposite sign to its value without QK-norm or is at most a third of it. Twelve runs without QK-norm, all at 8.5×10^{-3} or above, could not be run (§6.4). If they had run and diverged, as all 36 finished runs at 12×10^{-3} and above did, both conditions would hold, and by this rule QK-norm would remove the tolerance. Of the 12 finished runs at 8.5×10^{-3} , 7 ended at losses of 5.3 to 6.4, and of the three runs missing there, one could change the verdict. Had global $h = 4$ on seed 2 ended between 5.7 and 6.9, the second condition would fail for global $h = 4$, and the rule would not count the tolerance as removed. With QK-norm the heads also trail their parameter-matched baselines at all nine learning rates, global $h = 4$ trailing the control by 0.011–0.052 and local $h = 4$ trailing mHC by 0.006–0.066.

One difference in favor of the heads survives QK-norm, and it is in the gradient norm. Papers on the HC family show gradient-norm curves as evidence of stability [18, 22, 40, 42, 48], and in that measure the heads differ from mHC (Figure 5, bottom). Without QK-norm and at 4.2×10^{-3} , the gradient norm before clipping stays below 0.9 in the last 80% of training for both head variants on

every seed and below 1.6 for the residual, while it reaches 6–16 for mHC and 6–20 for the control. At 6×10^{-3} , where every variant fails, all of them cross the clipping threshold on 29–84% of those steps. With QK-norm the difference extends over the whole grid above the optimum of mHC, 1.5×10^{-3} . At that optimum mHC crosses the threshold on 0.03% of the late steps, on 0.1% at 2.1×10^{-3} , which is still within 0.01 of its best, and on 0.3–3% from 3×10^{-3} upward. The control behaves in the same way. Global $h = 4$ crosses on almost none of the late steps up to 8.5×10^{-3} and on 0.4% above it, local $h = 4$ on at most 0.15% except at 12×10^{-3} (0.9%), and the residual on at most 0.1%. Paired by seed, both head variants cross less often than mHC at every learning rate from 2.1×10^{-3} to 24×10^{-3} and on every seed. These are small counts, since 0.1% of the 4800 late steps is five steps.

The difference does not reach the loss. No run with QK-norm has a loss spike, clipping caps the gradient norm at 1.0, and with QK-norm, mHC ends below both head variants at every learning rate. Nor do the heads add a smoothness that the residual lacks. They remove a roughness that mHC adds at 27M, although the authors of mHC report a gradient norm as steady as that of the residual at 27B [40].

At 112M our data cannot settle the question. The 112M runs are without QK-norm and log every 25th step, which is too sparse to see a rate like mHC’s 0.1% at 27M. From 0.53×10^{-3} to 1.5×10^{-3} on TPU v6e, no variant crosses on more than 1% of the logged late steps, and the residual, which almost never crosses at 27M, crosses most often at three of these four learning rates. At 2.1×10^{-3} , 2.8 times the optimum, where every variant run there has failed, the heads cross on 28–38% and mHC on 85%, as at 27M and 6×10^{-3} . A repeat at 112M with QK-norm and per-step logs could not be run (§6.4).

Table 7: The stability test at 27M (TPU v6e, seeds 0–2). The column *best LR* gives the best learning rate of each variant on the grid ($\times 10^{-3}$), *max logit* the largest attention logit over layers at 6×10^{-3} and step 1000, averaged over seeds, and *clipped* the share (%) of steps in the last 80% of training whose gradient norm before clipping exceeds 1.0, at 4.2×10^{-3} . ΔS is the difference in the learning-rate sensitivity of Wortsman et al. [38], paired by seed and negative when the first variant is less sensitive. A diverged run counts as ln 16384. Without QK-norm the full grid is complete only on the seeds shown, since twelve runs at 8.5×10^{-3} and above could not be run. * marks established effects.

	without QK-norm			with QK-norm		
variant	best LR	max logit	clipped (%)	best LR	max logit	clipped (%)
mHC	2.1	2069	0.39	1.5	20.4	0.47
global $h = 4$	2.1	1926	0.00	3	18.7	0.00
local $h = 4$	3	2004	0.00	4.2	19.1	0.00
MLP control	2.1	2710	0.68	2.1	20.4	0.49
residual	2.1	2474	0.01	1.5	19.9	0.00
ΔS , all nine learning rates						
global – mHC	−0.013, +0.472 (seeds 0, 1)			+0.015 ± 0.002* (0/3)		
local – mHC	−0.067, −0.038 (seeds 0, 1)			+0.006 ± 0.003* (0/3)		
global – control	−0.102 (seed 0)			+0.011 ± 0.007 (0/3)		
ΔS , 1.5×10^{-3} to 6×10^{-3}						
global – mHC	−0.018 ± 0.013 (2/3)			+0.004 ± 0.001* (0/3)		
local – mHC	−0.028 ± 0.010* (3/3)			+0.000 ± 0.005 (2/3)		
global – control	−0.048 ± 0.012* (3/3)			+0.007 ± 0.005 (1/3)		

5.6 Ablation of the per-head blocks

Keeping only some of the three blocks per head separates their roles (Figure 6 and Table 8). These runs were planned at 3×10^{-3} , the rate of the v5e experiments, before the grid showed that on v6e this rate lies above the optimum of mHC, global $h = 4$ and the control. We added the optimum, 2.1×10^{-3} , after two of the three seeds at 3×10^{-3} had finished, and ran it regardless of what they showed. A third rate, 6×10^{-3} , tests which blocks carry the tolerance of §5.4.

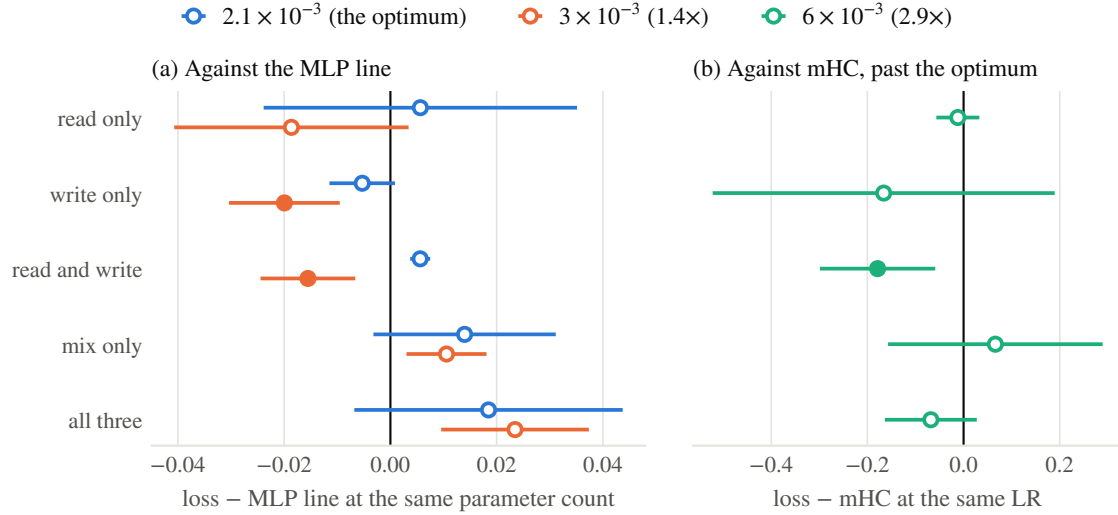


Figure 6: Ablation of the per-head blocks (global predictor, $h = 4$, $D = 384$, TPU v6e, seeds 0–2, ± 2 SE). Filled markers show variants that beat the MLP line by more than 2 SE with every seed below it in (a), and established effects in (b). (a) At the optimum no block beats what its parameters buy in the feed-forward layer, while at 1.4 times the optimum the write alone and the read and write together do. (b) At 2.9 times the optimum only read and write per head with a shared mix keep a lead over mHC on every seed.

Table 8: Global $h = 4$ with only some blocks per head ($D = 384$, TPU v6e, seeds 0–2). The table gives the paired difference to mHC at each learning rate and, at the two lower ones, the excess over the MLP line at the parameter count of the variant, which is negative when the variant uses its parameters better than the feed-forward layer. * marks established effects.

per head	params	2.1×10^{-3} (optimum)		3×10^{-3}		6×10^{-3}
		Δ vs. mHC	excess	Δ vs. mHC	excess	Δ vs. mHC
read	27.66M	$+0.002 \pm 0.015$	+0.006	$-0.025 \pm 0.011^*$	-0.019	-0.012 ± 0.022
write	27.64M	$-0.009 \pm 0.003^*$	-0.005	$-0.026 \pm 0.005^*$	-0.020	-0.166 ± 0.178
read and write	27.95M	-0.002 ± 0.001	+0.006	$-0.029 \pm 0.004^*$	-0.016	$-0.179 \pm 0.060^*$
mix	28.52M	-0.001 ± 0.009	+0.014	$-0.015 \pm 0.004^*$	+0.011	$+0.066 \pm 0.112$
all three (global $h = 4$)	29.13M	-0.003 ± 0.013	+0.018	$-0.015 \pm 0.007^*$	+0.023	-0.068 ± 0.048

At 3×10^{-3} every split that keeps the mix shared beats mHC on every seed, by 0.025–0.029, and lies 0.016–0.020 below the MLP line, the read alone by less than twice its standard error, so the few extra parameters of these splits appear better spent than in the feed-forward layer. The per-head mix alone costs 1.2M parameters, nearly four times as many as the read or the write, and buys less than those parameters would in the feed-forward layer. At 6×10^{-3} , per-head read and write with a shared mix lead mHC by 2.6 times as much as the full split does (-0.179 ± 0.060 , 3 of 3 seeds), and in the last 80% of training their gradient norm, logged every 25th step, peaks lower than that

of any other variant at that rate in the grid without QK-norm (5 on average over seeds, against 16 for the full split and 51 for mHC). The per-head mix alone keeps none of the lead and peaks highest of the split variants (93).

At the optimum the gains seen at 3×10^{-3} are gone. No variant lies below the MLP line by more than twice its standard error. Every one trails the MLP control of global $h = 4$, which has the parameter count of the full split, on every seed, by 0.013–0.023, and the per-head read alone, which is the group-wise readout of uHC but with a global predictor, is $+0.002 \pm 0.015$ against mHC. The write alone comes closest. It beats mHC on every seed by 0.009 ± 0.003 , but the MLP line gives -0.004 at its parameter count, and the remaining excess of -0.005 falls short of $2 \text{ SE} = 0.006$. The gains at 3×10^{-3} most likely came from the same tolerance of a too-high learning rate that the full split shows, and they do not reflect a better use of parameters.

A rule fixed before these results sent the read alone and the read and write together to $D = 768$ if they beat the MLP line at 3×10^{-3} by more than twice the standard error and on every seed. The read alone missed this bar (excess -0.019 against $2 \text{ SE} = 0.022$), and read and write together met it. At $D = 768$ and 0.75×10^{-3} on TPU v6e, paired by seed, the variant with read and write per head is -0.006 ± 0.004 against mHC (3 of 5 seeds), an excess over the MLP line at its parameter count of -0.002 against $2 \text{ SE} = 0.008$, with 2 of 5 seeds below the line. Against the MLP control of global $h = 4$ it differs by $+0.005 \pm 0.007$. The rule did not cover the write alone, which was therefore not run at $D = 768$.

6 Discussion and Limitations

6.1 Comparing connection variants

At the pilot learning rate of mHC, global $h = 4$ looked like a clear improvement, 0.019 better than mHC on 9 of 10 seeds and about 40% of the gain of mHC over the residual. Two controls removed this gain. The same parameters in a wider feed-forward layer gain as much, and with every variant at its own best learning rate the lead disappears at both widths. What remained was an advantage above the optimum, which describes how training fails and says little about the model one would train, and at 27M QK-norm removes it. Connection variants are often compared at one shared learning rate and at unequal parameter counts. Our results suggest that both controls are needed before a gain in this family is read as structural.

6.2 Stability at larger scale

A gentler failure above the optimum would be valuable if it predicted fewer instabilities in large models, where a failure costs a whole run. Two lines of work make this argument plausible in general. Wortsman et al. [38] reproduce in small models, by raising the learning rate, two instabilities first seen in large models, and they measure the robustness of a model by its learning-rate sensitivity, the expected excess loss over a learning-rate grid. Qwen3.8-Next accepts a new recipe only if a 28-layer mixture-of-experts model with 25B parameters (3B active) is at least as stable as the structure it replaces at two to four times its optimal learning rate, counted in loss spikes and clip crossings, and the stability found there held in the 125B-parameter production model [30]. For four reasons, our results do not support the same argument for multi-head mHC.

First, the failure we observe is one that QK-norm removes. Above the optimum our 27M models fail through attention-logit growth, the first of the two instabilities of Wortsman et al. [38], and the heads do not prevent it (§5.4). QK-norm prevents it at every learning rate we tried, and with QK-norm the tolerance of the heads is gone (§5.5). Current open models use QK-norm [10, 35, 41], as Wortsman et al. do by default, so the property we measured belongs to a recipe without this protection.

Second, the heads change how much the loss rises above the optimum but leave unchanged the learning rate at which models fail (§5.4). Wortsman et al. show that this learning rate falls as models grow when attention logits are not normalized, and they predict it by extrapolating the largest attention logit. At 27M without QK-norm, mHC, both head variants and the residual first lose more than 0.1 at 6×10^{-3} , and the control at 4.2×10^{-3} . With QK-norm, all of them except the residual first do so at 17×10^{-3} . Wortsman et al. also remark that in practice a lower sensitivity matters little when the optimal learning rate does not change. Without QK-norm, the optimum with heads stays within one grid point of that of mHC at both widths (§5.1).

Third, there are no loss spikes for the heads to reduce. Wortsman et al. study slow divergence and leave loss spikes out of scope, tying them to the optimizer through the second-moment decay rate of Adam. Wherever training does not fail, none of the 27M runs with per-step logs has a loss spike by the measure of OLMo 2 (§5.4). The spikes that matter at scale may also have no small-scale counterpart. The Adam instability behind the prominent loss spikes of a 546B model did not appear at 7B and 30B [24].

Fourth, the precedents differ from our case. In Qwen3.8-Next and in gated attention [29], the more stable design also has the lower loss in the comparisons reported, whereas multi-head mHC gives no gain at equal parameters. The stress test of Qwen3.8-Next runs at 25B parameters, compares

against a recipe already proven at scale and was confirmed in production, whereas ours runs at 27M. The authors of Qwen3.8-Next identify a bounded elementwise gate on what each layer reads from the stream as a key contributor to the stability of their design. They drop the stream mix, which they see as a possible source of instability, and the mix is the part that multi-head mHC multiplies by h . At 7B and with QK-norm, OLMo 2 trained normally at two, three and four times its learning rate and was unstable only at ten times that rate [35]. Sensitivity to the training hyperparameters also shrinks as models grow [20], so a difference in sensitivity measured in small models may matter less in large ones.

The gradient norm is the one measure that favors the heads even with QK-norm. The gradient norm of the heads crosses the clipping threshold less often than that of mHC (§5.5), and gradient-norm curves are part of how the HC family argues for stability. We still cannot claim this as a benefit. What the heads remove at 27M is a roughness that mHC adds and the residual does not have, so at most they restore the profile of the residual, which the authors of mHC report for mHC at 27B. The roughness costs mHC nothing we can measure, since with QK-norm, mHC has no loss spikes and a lower loss than the heads at every learning rate. Near the optimum the difference does not show at 112M, the one larger model we have, although there we only have runs without QK-norm, logged too sparsely to detect crossings as rare as those of mHC at 27M. To our knowledge, nobody has tested whether clip crossings in a small model foretell loss spikes in a large one, and Qwen3.8-Next counts them in the large model itself.

What we can report is a gentler failure above the optimum and a smoother gradient norm. Neither is a stability benefit we can claim for larger models. Testing either would require a model large enough to fail at its own tuned learning rate, or a stress test at the scale of the one used for Qwen3.8-Next.

6.3 Why the heads do not help

Head splits have helped in other connections. The same split applied to the depth read of Attention Residuals beats a single query on web data at 100M–1B parameters with the learning rate tuned for each method [21], although an anonymous submission with the same title attributes most of the gain of the split over a single query to the flatter softmax of smaller heads and only a small part to routing each subspace separately [2], and the authors of Attention Residuals found their own per-head version slightly worse [14]. Per-channel reads helped in the residual ablations of Qwen3.8-Next, which use mixture-of-experts models with 25B parameters (3B active) [30], and per-channel reads and writes in diffusion transformers whose streams hold different subpatches [17]. In mHC the copies start identical, and a trained model uses only some of them at each site [1, 47], so at our scale a per-group choice among the copies may have little to choose from. Consistent with this, the per-group read alone is $+0.002 \pm 0.015$ against mHC at the optimum.

The per-group doubly stochastic mix is what separates our design from earlier and concurrent work, and it is also the block that contributes least. It lies above the MLP line at both learning rates where we drew the line, at 6×10^{-3} it keeps none of the lead of the full split over mHC, and it has the largest late gradient norm of the split variants there. This agrees with the recent designs that drop the stream mix [9, 17, 30] and with the finding that read and write gates alone match mHC [4].

Within mHC the tolerance needs the split, since a faster connection gives mHC none of it, and it sits in the per-group read and write. It is not resistance to the failure itself, because attention logits grow almost as fast with heads as without them. A plausible explanation, which we did not test, is that per-group coefficients let the connection turn down the channel groups whose

updates grow too large, which a single scalar per copy cannot do without silencing every channel at once. The one attention diagnostic in which the heads differ, a first attention layer that stays almost uniform, is shared with the residual. At 27M the residual is also less sensitive than mHC over the five learning rates up to 6×10^{-3} (-0.019 ± 0.006 , 3 of 3 seeds), so this layer may be part of the explanation, but we did not test whether it causes the tolerance.

6.4 Limitations

Our models are small. They have 27M and 112M parameters and train on 98M tokens, or 3.6 and 0.9 tokens per parameter, far short of compute-optimal training. The 393M-token runs, at one learning rate, with three seeds and with each token seen about twice, are our only test of training length. The gain of Multi-Head Attention Residuals over the standard residual was larger at 350M and 1B parameters than at 100M [21], and a per-channel residual gate first became significant at 590M parameters [16], so a gain at larger scale is not excluded.

The coverage of the study is also limited. All runs use one dataset, one tokenizer and contiguous channel groups, and all but one variant use $n = 4$ copies. Only mHC, global and local $h = 4$, the control and the residual have a learning-rate grid. The block ablations were run at three learning rates and the other variants of Table 5 at one. We tune only the learning rate, keep the rest of the mHC recipe and measure only validation loss, without downstream evaluations. The control widens the feed-forward layer, and another use of the same parameters, such as wider attention or more layers, could buy more or less.

Most comparisons rest on 3 to 10 seeds. The standard error of the difference between global $h = 4$ and its control is 0.007 at 27M (10 seeds at 3×10^{-3}) and 0.006 at 112M (7 seeds), so structural effects smaller than about 0.015 and 0.012 could go undetected. With about twenty variants and several learning rates per width, a few chance effects are expected, and one appeared and then vanished (local $h = 2$ at -0.018 on 4 seeds and -0.003 on 10). The comparisons that carry the conclusion are either far from the threshold or repeated on a second chip.

The stability test runs at 27M only, with three seeds per learning rate, one batch size, AdamW with $\beta_2 = 0.95$, clipping at 1.0 and a 6000-step cosine schedule. It therefore cannot reproduce instabilities that appear only in long or large runs, or under a constant learning rate as in the stress test of Qwen3.8-Next. Twelve of its 240 planned runs, all without QK-norm at 8.5×10^{-3} or above and all but one on seed 2, could not be run for lack of spot TPU capacity. Every finished run in that range diverged except seven at 8.5×10^{-3} , and one missing run at that rate could have changed the pre-registered verdict (§5.5). A repeat at 112M with QK-norm and per-step logs, planned to test the gradient-norm difference and the comparison at equal parameters, could not be run either, because spot TPU capacity was too scarce and the VMs that did start were preempted before a run could finish. With QK-norm, mHC and the residual are best at 1.5×10^{-3} , the lowest point of the grid, so their true optimum may lie lower. That would only widen the lead of mHC over local heads at the best learning rates. It would raise the sensitivity of mHC by as much as its best loss falls. Since the loss of mHC changes by less than 0.01 from 1.5×10^{-3} to 3×10^{-3} , that amount is likely small, although it could be as large as the 0.006 by which local $h = 4$ is more sensitive than mHC.

Our results come from two TPU generations whose runs are not interchangeable (Appendix A). Each comparison stays on one of them, and where both ran the same comparison we report both. Finally, we tested the group-wise read of the closest concurrent work [5] only with a global predictor and without its low-rank write-back.

7 Conclusion

At 27M and 112M parameters, with every variant at its own best learning rate and against parameter-matched controls, multi-head mHC gives no gain at equal parameters. With a global predictor it gains at most what its extra parameters gain in the feed-forward layer, and at the parameter count of mHC it gains nothing. At 27M the comparison comes out the same with QK-norm and, at one learning rate, with four times the tokens, and at the best learning rate of mHC no subset of the per-head blocks that we tested uses its parameters better than the feed-forward layer.

Without QK-norm, multi-head mHC fails more gently at too-high learning rates. At 27M this tolerance sits in the per-group read and write, and the failure is attention-logit growth, which the heads do not prevent. QK-norm removes it and makes the heads more sensitive to the learning rate than mHC, and in the stability test only a run that failed had loss spikes, so we cannot offer the tolerance as a stability benefit for larger models. The one difference in favor of the heads that survives QK-norm is a smoother gradient norm, which does not reach the loss at 27M and which we could not test with QK-norm at 112M. Whether it matters at scale remains open. So does whether a gain at equal parameters appears in larger models, where other connection variants have gained more than in small ones.

References

- [1] E. Alimaskina, G. Molodtsov, and A. Beznosikov. Analyzing Stream Collapse in Hyper-Connections: From Diagnosis to Mitigation, 2026. ICML 2026 Workshop on Weight-Space Symmetries, [arXiv:2606.03483](#).
- [2] Anonymous. Multi-Head Attention Residuals, 2026. Submitted to ICLR 2027, [OpenReview](#).
- [3] Anonymous. One Marginal Is Enough: What Doubly Stochastic Routing Buys Hyper-Connections, and What It Costs, 2026. Submitted to ICLR 2027, [OpenReview](#).
- [4] Anonymous. On the Design of Hyper-Connections: Simplification, Stability, and Scalability, 2026. Submitted to ICLR 2027, [OpenReview](#).
- [5] Anonymous. uHC: Uncoupled Hyper-Connections, 2026. Submitted to ICLR 2027, [OpenReview](#).
- [6] ByteDance Seed. Virtual Width Networks, 2025. [arXiv:2511.11238](#).
- [7] T. Dandachi and S. Diggs-Galligan. go-mHC: Direct Parameterization of Manifold-Constrained Hyper-Connections via Generalized Orthostochastic Matrices, 2026. [arXiv:2604.02309](#).
- [8] M. Dehghani, J. Djolonga, B. Mustafa, et al. Scaling Vision Transformers to 22 Billion Parameters, 2023. ICML 2023, [arXiv:2302.05442](#).
- [9] C. Ge, H. Jiang, T. Wang, et al. Chimera: Designing and Chinchilla-Scaling Hybrid Visual Diffusion Transformers, 2026. [arXiv:2607.28611](#).
- [10] Gemma Team, A. Kamath, J. Ferret, et al. Gemma 3 Technical Report, 2025. [arXiv:2503.19786](#).
- [11] Y. Gu, W. Zhou, H. Xing, and D. Mandic. TEMPER: Tensorized Efficient Manifold-Constrained Parameterization for Expressive Residual Routing, 2026. [arXiv:2608.07851](#).
- [12] H. Guo, X. Chen, B. Ke, Y. Lei, Z. Xu, S. Feng, Ximen, and W. Luo. oHC: Orthogonal Hyper-Connections on SO(4) via Quaternions, 2026. [arXiv:2609.02672](#).
- [13] A. Henry, P. R. Dachapally, S. S. Pawar, and Y. Chen. Query-Key Normalization for Transformers, 2020. Findings of EMNLP 2020, [arXiv:2010.04245](#).
- [14] Kimi Team. Attention Residuals, 2026. [arXiv:2603.15031](#).
- [15] Kimi Team. Kimi K3: Open Frontier Intelligence, 2026. [arXiv:2607.24653](#).
- [16] K. Kramer. Review Residuals: Update-Conditioned Residual Gating for Transformers, 2026. [arXiv:2606.31859](#).
- [17] T. Liang, S. Kou, Z. Xi, et al. Residual-Stream Burden Shapes Representation Learning in Diffusion Transformers, 2026. [arXiv:2609.33895](#).
- [18] Z. Liu, H. Zhang, and A. Li. Spectral-Sphere-Constrained Hyper-Connections, 2026. NeurIPS 2026, [arXiv:2603.20896](#).
- [19] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization, 2019. ICLR 2019, [arXiv:1711.05101](#).
- [20] N. Lourie, K. Cho, K. Ullrich, and S. Lotfi. Small-Scale Experiments: Are We There Yet?, 2026. [arXiv:2608.11859](#).
- [21] C. Luo, Z. Cai, and J. Hu. Multi-Head Attention Residuals, 2026. [arXiv:2607.27230](#).
- [22] A. Lyubinin. TBP-mHC: Full Expressivity for Manifold-Constrained Hyper Connections through Transportation Polytopes, 2026. [arXiv:2605.21724](#).
- [23] G. Melis, C. Dyer, and P. Blunsom. On the State of the Art of Evaluation in Neural Language Models, 2018. ICLR 2018, [arXiv:1707.05589](#).
- [24] I. Molybog, P. Albert, M. Chen, et al. A Theory on Adam Instability in Large-Scale Machine Learning, 2023. [arXiv:2304.09871](#).
- [25] A. Mutlu, Ş. Doğan, and T. Tuncer. mHC-SSM: Manifold-Constrained Hyper-Connections for State Space Language Models with Stream-Specialized Adapters, 2026. [arXiv:2605.08300](#).

- [26] S. Narang, H. W. Chung, Y. Tay, et al. Do Transformer Modifications Transfer Across Implementations and Applications?, 2021. EMNLP 2021, [arXiv:2102.11972](#).
- [27] M. Pagliardini, A. Mohtashami, F. Fleuret, and M. Jaggi. DenseFormer: Enhancing Information Flow in Transformers via Depth Weighted Averaging, 2024. NeurIPS 2024, [arXiv:2402.02622](#).
- [28] G. Penedo, H. Kydlíček, L. Ben Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. von Werra, and T. Wolf. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, 2024. NeurIPS 2024, [arXiv:2406.17557](#).
- [29] Z. Qiu, Z. Wang, B. Zheng, et al. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free, 2025. NeurIPS 2025, [arXiv:2505.06708](#).
- [30] Z. Qiu, Z. Wang, X. Li, et al. On the Design of Qwen3.8-Next Architecture: Evaluation, Efficiency, and Training Stability, 2026. [arXiv:2608.30320](#).
- [31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners, 2019. OpenAI technical report.
- [32] R. Sennrich, B. Haddow, and A. Birch. Neural Machine Translation of Rare Words with Subword Units, 2016. ACL 2016, [arXiv:1508.07909](#).
- [33] N. Shazeer. GLU Variants Improve Transformer, 2020. [arXiv:2002.05202](#).
- [34] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2021. Neurocomputing 2024, [arXiv:2104.09864](#).
- [35] Team OLMo, P. Walsh, L. Soldaini, et al. 2 OLMo 2 Furious, 2024. Shorter version at COLM 2025, [arXiv:2501.00656](#).
- [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention Is All You Need, 2017. NeurIPS 2017, [arXiv:1706.03762](#).
- [37] K. Wang, Y. Huang, J. Kang, D. González-Martínez, W. Liu, and S. Liu. DepthBench: Measuring How Residual Connections Enable More Computational Depth, 2026. [arXiv:2609.32534](#).
- [38] M. Wortsman, P. J. Liu, L. Xiao, et al. Small-Scale Proxies for Large-Scale Transformer Training Instabilities, 2023. ICLR 2024, [arXiv:2309.14322](#).
- [39] D. Xiao, Q. Meng, S. Li, and X. Yuan. MUDDFormer: Breaking Residual Bottlenecks in Transformers via Multiway Dynamic Dense Connections, 2025. ICML 2025, [arXiv:2502.12170](#).
- [40] Z. Xie, Y. Wei, H. Cao, et al. mHC: Manifold-Constrained Hyper-Connections, 2025. ICML 2026, [arXiv:2512.24880](#).
- [41] A. Yang, A. Li, B. Yang, et al. Qwen3 Technical Report, 2025. [arXiv:2505.09388](#).
- [42] Y. Yang and J. Gao. mHC-lite: You Don’t Need 20 Sinkhorn-Knopp Iterations, 2026. [arXiv:2601.05732](#).
- [43] B. Zeng, Y. Hao, Z. Wang, et al. Depth-Attention: Cross-Layer Value Mixing for Language Models, 2026. [arXiv:2606.05014](#).
- [44] S. Zhai, T. Likhomanenko, E. Littwin, et al. Stabilizing Transformer Training by Preventing Attention Entropy Collapse, 2023. ICML 2023, [arXiv:2303.06296](#).
- [45] B. Zhang and R. Sennrich. Root Mean Square Layer Normalization, 2019. NeurIPS 2019, [arXiv:1910.07467](#).
- [46] X. Zhang, X. Qin, S. Zou, et al. xHC: Expanded Hyper-Connections, 2026. [arXiv:2607.14530](#).
- [47] P. Zhao, X. Li, X. Yu, W. Guo, and Z. Dong. How Does mHC Use Its Residual Streams? Selective Routing and Near-Identity Mixing, 2026. [arXiv:2609.05309](#).
- [48] W. Zhou, Y. Gu, G. Iacovides, and D. Mandic. KromHC: Manifold-Constrained Hyper-Connections with Kronecker-Product Residual Matrices, 2026. ICML 2026, [arXiv:2601.21579](#).
- [49] D. Zhu, H. Huang, Z. Huang, Y. Zeng, Y. Mao, B. Wu, Q. Min, and X. Zhou. Hyper-Connections, 2024. ICLR 2025, [arXiv:2409.19606](#).
- [50] D. Zhu, H. Huang, J. Zhou, Z. Huang, Y. Zeng, B. Wu, Q. Min, and X. Zhou. Frac-Connections: Fractional Extension of Hyper-Connections, 2025. [arXiv:2503.14125](#).

Appendix

A Comparability Across Chips

TPU v5e, first on Kaggle and later on Google Cloud, ran the experiments of Table 5, the 393M-token runs, a 112M grid with two seeds and a few 27M runs at other learning rates. On Google Cloud, a partial replay of a 27M run and a full replay of a 112M Kaggle run were bit-identical at every logged step. All other TPU runs used spot TPU v6e. Before reading any v6e result, we repeated on v6e the runs on which the $D = 384$ and $D = 768$ conclusions rested, with pass criteria fixed beforehand. Each run had to fall within the seed-to-seed range of its variant on v5e, and each paired gap had to agree in sign and lie within the v5e standard error.

Table 9: The same comparisons on the two chips. Paired gaps at $D = 384$ and 3×10^{-3} , given as mean $\pm$ SE and the number of seeds on which the first model is better. * marks established effects.

gap	v5e, all seeds	v5e, seeds 0–2	v6e, seeds 0–2
global $h = 4$ – mHC	-0.019 ± 0.006 (9/10)	$-0.024 \pm 0.005^*$ (3/3)	$-0.015 \pm 0.007^*$ (3/3)
global $h = 4$ – MLP control	$+0.005 \pm 0.007$ (2/10)	$+0.017 \pm 0.024$ (1/3)	$+0.023 \pm 0.008^*$ (0/3)
MLP control – mHC	-0.024 ± 0.009 (8/10)	-0.041 ± 0.024 (2/3)	$-0.038 \pm 0.009^*$ (3/3)
local $h = 4$ – mHC	$+0.003 \pm 0.008$ (3/6)	-0.006 ± 0.008 (2/3)	-0.018 ± 0.012 (2/3)
mHC – residual	$-0.045 \pm 0.006^*$ (4/4)	$-0.041 \pm 0.007^*$ (3/3)	$-0.041 \pm 0.004^*$ (3/3)

Runs on v6e are deterministic. A v6e run repeated on another single-chip VM, or on one chip of an 8-chip VM, is identical to the original at every logged step and evaluation, so a result does not depend on the VM that produced it.

Near the optimum, at 3×10^{-3} ($D = 384$, seeds 0–2), every gap has the same sign on both chips (Table 9), but only two of five agree within the standard error over all v5e seeds, with misses of 0.014–0.021. Over all v5e seeds, the gap between local $h = 4$ and mHC even has the opposite sign. Per run, the v6e loss minus the v5e loss averages $+0.007$ with a standard deviation of 0.014, as large as the seed-to-seed spread of a variant on one chip (0.012–0.018). A v6e run starts from the same weights and sees the same batches as its v5e twin, but floating-point differences send its trajectory elsewhere, so it behaves like a new seed. Against the combined standard error of two independent estimates, no gap differs between the chips by more than 1.7 SE.

At 6×10^{-3} ($D = 384$, seeds 0–1), where every variant fails, the loss rises sharply early in training on both chips and recovers as the learning rate decays. The final loss depends on when it recovers, and the same seed lands up to 0.21 apart on the two chips. Both head variants stay ahead of mHC and the control on average, by 0.05–0.17 on v6e and 0.19–0.29 on v5e. Over the late steps logged every 25th, the largest gradient norm, averaged over seeds, is 14 and 9 for global and local $h = 4$ against 69 for mHC and 559 for the control on v6e, and 7, 13, 24 and 97 on v5e.

At 112M and 1.5×10^{-3} ($D = 768$, seeds 0–1), global $h = 4$ leads mHC by 0.068 on v5e and 0.051 on v6e, but its lead over the control shrinks from 0.034 ± 0.013 (2 of 2 seeds) to 0.011 ± 0.032 (1 of 2).

As written, the check therefore fails, because its criteria assumed that a v6e run would track its v5e twin. For this reason we never put numbers from the two chips into one comparison. Every experiment runs on one chip type with its own baselines, and where both chips ran the same comparison, the two estimates are pooled only as independent ones.

B Learning-Rate Grids

Tables 10 and 11 give the mean final validation loss at every point of the grids behind Figure 3 and Table 4. Bold marks the best learning rate of each variant under the rule of §4.2.

Table 10: Learning-rate grid at 27M ($D = 384$, TPU v6e). Mean final validation loss over seeds 0–2. A dash marks a point not run in this grid.

model	1.5×10^{-3}	2.1×10^{-3}	3×10^{-3}	4.2×10^{-3}	6×10^{-3}
mHC	4.1347	4.1215	4.1561	4.1867	5.0481
global $h = 4$	4.1372	4.1180	4.1409	4.1617	4.9800
MLP control	4.1108	4.0998	4.1179	4.2066	5.1531
local $h = 4$	4.1651	4.1411	4.1380	4.1984	4.9464
residual	4.1933	4.1932	4.1971	4.2528	–

Table 11: Learning-rate grid at 112M ($D = 768$). Mean final validation loss over seeds 0–2 on TPU v6e and over seeds 0–1 on TPU v5e. Table 4 adds seeds 3 and 4 on v6e at the best learning rate of every variant except the residual. A dash marks a point not run.

chip	model	0.53×10^{-3}	0.75×10^{-3}	1.06×10^{-3}	1.5×10^{-3}	2.1×10^{-3}
v6e	mHC	3.9667	3.9458	3.9501	4.0952	4.7830
	global $h = 4$	3.9593	3.9429	3.9422	4.0319	4.6719
	MLP control	3.9599	3.9378	3.9492	4.0862	4.7860
	local $h = 4$	3.9844	3.9435	3.9644	4.0689	4.6610
	residual	4.0230	4.0035	4.0019	4.1557	–
v5e	mHC	3.9781	3.9292	3.9593	4.1097	4.7487
	global $h = 4$	3.9651	3.9367	3.9222	4.0418	4.7792
	MLP control	3.9574	3.9387	3.9472	4.0754	4.7240
	local $h = 4$	3.9914	3.9517	3.9650	4.0405	4.7523
	residual	–	–	–	4.1669	–

C Implementation and Compute

A TPU pads the last two axes of every array to 8×128 tiles. In the natural token-major layout, the $n \times n$ mixes were padded 64-fold and models with heads never finished compiling. We therefore store the stream as $[n, D, \text{tokens}]$, with copies and heads on the leading axes. In this layout every head count runs at the speed of mHC.

We found a compiler fault on v6e. When two accumulated micro-batches were fused into one XLA graph with gradient clipping and the AdamW update, the gradients were wrong, and the gradient norm at step 0 was 8.43 instead of 4.39. Every v6e run therefore takes its 16-sequence batch in one micro-step, which changes the gradient only at the level of floating-point rounding.

The study comprises 629 TPU runs. Of these, 603 follow the 98M-token schedule, with 117 on Kaggle v5e, 39 on Google Cloud v5e and 447 on Google Cloud v6e. The v6e runs include the 228 runs of the stability test of §5.4 and §5.5, 42 of which replay earlier runs. Another 15 runs use 393M tokens, and 11 short runs on Kaggle measure compile time and throughput. A pilot added 19 shorter GPU runs. The Google Cloud runs cost \$206 on spot VMs, of which \$77 went to the stability test and to the 112M repeat that could not be completed. Preempted runs restart from step 0, and since runs are deterministic on a given chip type, a restart changes the cost but not the result.

The code, the configuration, metrics and final summary of every run, the decision rules with their timestamps, and the scripts that produce every number and figure in this report are available at <https://github.com/marcoshernanz/multi-head-mhc>.