Data-Adaptive Mahalanobis Metric Learning for Cross-Head Attention in Transformers

Zhenke Duan[1,2] **Xin Li**[1] **Jiqun Pan**[1]
Dong Xiaofei[1] **Xinyuan Song**[3] **Hanwen Ning**[1,4*]

[1] School of Statistics and Mathematics,
Zhongnan University of Economics and Law, Wuhan, China
[2] CodeSoul.Co, Hangzhou, China
[3] Department of Statistics,
The Chinese University of Hong Kong, Hong Kong SAR, China
[4] Innovation and Talent Base for Digital Technology and Finance, ZUEL

Abstract

In this paper, we propose Data-Adaptive Mahalanobis Cross-Head Attention (DAMCHA), a simple and effective attention mechanism that generalizes Multi-Head Attention (MHA) through the lens of metric learning. Standard MHA implicitly learns a static, head-wise Euclidean similarity corresponding to the diagonal blocks of a unified cross-head metric matrix, discarding all off-diagonal interactions across heads. DAMCHA addresses this limitation via two key innovations. First, it learns the full cross-head metric matrix, inducing a Mahalanobis-like similarity that explicitly captures inter-head interactions beyond the restricted block-diagonal structure of MHA. Second, it parameterizes the metric matrix as an input-dependent function, enabling the attention geometry to dynamically adapt to the intrinsic manifold structure of the data rather than remaining fixed after training. We further introduce principled regularization strategies, most notably stack-wise parameter sharing, to ensure computational efficiency and stable optimization. DAMCHA serves as a drop-in replacement for MHA and its variants in various Transformer architectures, delivering superior expressive power and learning performance with comparable or reduced computational overhead. Extensive experiments demonstrate that DAMCHA-based Transformers consistently outperform competing baselines across a range of benchmark tasks. Code is available at https://github.com/erwinmsmith/DAMCHA.

1 Introduction

MHA is a core block of Transformer architecture [1, 2, 3]. MHA essentially learns a set of similarity matrices induced by a bilinear form (usually denoted as $M_{i,i} = W_Q^{(i)}(W_K^{(i)})^\top$) through a query–key mechanism [4, 5, 6]. These learned head-wise metrics are used to compute attention scores, which capture token matching intensities across disparate representation subspaces [2, 7, 8]. This formulation naturally aligns MHA with metric learning, where discriminative similarity structures are learned to support downstream tasks.[9, 10, 11, 12, 13].

Despite its extensive successes in diverse domains [14, 15, 16], when examined through this lens, standard MHA exhibits several inherent structural deficiencies that limit its expressive power. First, it

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

assumes conditional independence between heads for a given input, where similarities are defined solely within each head and merely concatenated at the output. This ignores the cross-head synergies and higher-order interactions corresponding to $M_{i,j} = W_Q^{(i)}(W_K^{(j)})^\top$ for $i \neq j$, making it difficult to represent coupled geometries between complex feature subspaces [17, 18, 19]. Second, the similarity metrics $M_{i,i}$ remain static post-training, which implies that, during inference, the attention cannot adapt to the input data, failing to match the non-linear and locally anisotropic manifold structures prevalent in real-world data. In addition, because $M_{i,i}$ relies on a factorized formulation, its rank is inherently capped by the projection dimension, introducing a well-known low-rank bottleneck that limits representation capacity [20, 21, 22]. These issues are especially problematic in tasks where the geometry of the data distribution is critical [23, 24].

In this paper, we propose a paradigm shift: rather than viewing the attention as a collection of independent, dot-product, head-wise matrices, it should be understood as a single, learnable Mahalanobis-like metric defined over the joint head space (Here, the term Mahalanobis-like is used to emphasize that our formulation can fully exploit off-diagonal blocks; we do not require the resulting Mahalanobis-like metric matrix to be positive definite). In contrast, standard MHA corresponds to a restrictive block-diagonal constraint that discards all off-diagonal couplings, thereby enforcing an effectively Euclidean similarity measure. From a learning perspective, MHA learns only a global, input-independent similarity matrix that approximates an average attention geometry across the dataset. Guided by this high-level interpretation, we propose a novel attention mechanism, termed DAMCHA. DAMCHA introduces two primary innovations. *Structural Expansion*: It eliminates structural independence by explicitly learning a full cross-head metric. While its diagonal blocks recover conventional within-head similarities, the off-diagonal blocks encode how different heads should cooperate, compensate, or modulate one another, expanding attention from an additive mixture of isolated similarities to a genuine Mahalanobis-like metric. *Dynamic Adaptivity*: Unlike MHA which keeps underlying parameters static, DAMCHA parameterizes the similarity matrix by a learnable function of the input. This allows the model to dynamically select appropriate geometries for different contexts, aligning the attention operator with the local structure of the data manifold. Several prior works have sought to improve upon MHA through head interaction or geometric enrichment, including THA [25], DCMHA [19], ColMHA [26], MoA [27], and MMA [28]. However, none of these approaches is grounded in metric learning, and none jointly addresses cross-head interaction and input-adaptive metric learning within a unified framework.

Practically, DAMCHA yields benefits beyond expressiveness: (i) improved modeling of complex dependencies spanning multiple semantic factors (e.g., content–position or syntax–semantics); (ii) enhanced adaptability under distributional shifts or heterogeneous inputs, as the attention geometry reconfigures itself to match the current context; (iii) by directly learning the similarity matrices $M_{i,j}$s through a functional mapping rather than the conventional linear query-key projections, DAMCHA naturally bypasses the low-rank bottleneck. We further introduce principled regularization strategies, most notably stack-wise parameter sharing, to control model complexity. Since the attention matrices in DAMCHA are generated by a compact metric learner, the overall parameter count remains comparable to standard MHA despite the expanded metric structure. This design ensures that the gains in expressiveness come at no significant cost in computational overhead or memory footprint.

Experimental results show that DAMCHA outperforms Transformers across a variety of architectures (e.g. GPT-2) and scales. In language modeling, DAMCHA is particularly superior. Notably, DAMCHA yields improvements in token-level pretraining; these gains are even stronger on mixed datasets, which are inherently cross-view. We further validate DAMCHA on autoregressive, decoder-only Transformers as well as Vision Transformers in generative and discriminative tasks. We provide extensive simulation experiments and ablation studies to elucidate the mechanisms of DAMCHA.

2 Related Work

The Transformer relies on self-attention [1], where multi-head attention projects queries and keys into parallel subspaces [2]. Empirical studies show heads often converge to similar or near-identical patterns [7, 29, 30], so adding heads does not guarantee more diverse representations [2]. This arises from the block-diagonal metric structure, which excludes cross-head interactions, inducing low-rank bottlenecks [21, 8] and redundant convergence [7]. Existing remedies, including cross-head interactions or output-level mixing and gating [17, 31, 19, 20, 32], operate post-hoc on attention

weights or outputs rather than similarity formation, modifying attention utilization without altering the underlying Euclidean dot-product metric and leaving the core structural limitation unresolved.

From a metric learning perspective, attention can be interpreted as a parameterized similarity metric [6], and existing metric-aware designs focus on domain-specific settings [33, 34, 35, 36], neglecting multi-head structural features. There is currently no unified framework that integrates Mahalanobis metric learning with multi-head attention to jointly model both intra- and cross-head similarities, leaving open the question of whether similarity metric should adapt dynamically to the input or remain fixed after training [23, 37]. For heterogeneous information interaction, recent work has also explored bidirectional attention to exchange complementary information across different feature domains, such as the spatial and Doppler representations in RaUF [38].

3 DAMCHA

3.1 Cross-Head Metric Matrix as a Conceptual Formulation

Let $X \in \mathbb{R}^{L\times d_{\text{model}}}$ denote an input sequence of length L, where $d = d_{\text{model}} = h\cdot d_h$, h is the number of attention heads, and d_h is the hidden dimension of each head. In standard multi-head attention (MHA), the i-th head computes query and key representations as $Q_i = XW_Q^{(i)}$ and $K_i = XW_K^{(i)}$, where $W_Q^{(i)}, W_K^{(i)} \in \mathbb{R}^{d_{\text{model}}\times d_h}$. The corresponding pre-softmax score matrix is

$$S_i^{\text{MHA}}(X) = \frac{1}{\sqrt{d_h}} Q_i K_i^\top = \frac{1}{\sqrt{d_h}} XW_Q^{(i)} \big(W_K^{(i)}\big)^\top X^\top. \tag{1}$$

Thus, from a metric-learning perspective, each head learns an independent bilinear similarity metric $M_{i,i} = W_Q^{(i)} \big(W_K^{(i)}\big)^\top$. Equivalently, standard MHA can be interpreted as a block-diagonal metric structure over the joint head space: $M_{\text{MHA}} = \text{diag}\big(M_{1,1}, M_{2,2}, \ldots, M_{h,h}\big)$. This block-diagonal structure implicitly assumes that the similarity geometry of each head is formed independently, without explicitly modeling metric-level interactions between different heads. To describe possible interactions beyond this restriction, we introduce the Cross-Head Metric Matrix (CHM):

$$M = \begin{pmatrix} M_{1,1} & \cdots & M_{1,h} \\ \vdots & \ddots & \vdots \\ M_{h,1} & \cdots & M_{h,h} \end{pmatrix}, \tag{2}$$

where each block $M_{i,j}$ represents a potential metric-level interaction between the query subspace associated with head i and the key subspace associated with head j. When all off-diagonal blocks vanish, CHM reduces to the block-diagonal metric structure of standard MHA. When off-diagonal blocks are allowed, attention is no longer restricted to independent head-wise similarity measurements.

3.2 Adaptive Learning of CHM

Let $X \in \mathbb{R}^{L\times d}$ denote the input token, and let each metric block satisfy $M_{i,j}(X) \in \mathbb{R}^{d_h\times d_h}$. In MHA, all $M_{i,i}$s are fixed after training, which implies that the same projection matrices are applied to all inputs X in inference process. Although attention score $XM_{i,i}X^\top$ vary with the input contents, the underlying metric $M_{i,i}$ itself is input-independent. In other words, MHA learns a static metric that is globally shared across all contexts. This static design may be suboptimal in practice. For instance, the relevance between two features may be strong in one context but weak in another [19, 39, 40, 41], yet a fixed metric cannot adapt its attention geometry accordingly.

To increase the expressiveness of the attention mechanism, we parameterize CHM as an input-dependent function. Concretely, a learnable forward network $f_\theta(\cdot)$ (e.g., an MLP or a hyper-network) is introduced to map X to an adaptive CHM, which follows

$$f_\theta(X) \to M(X) = \begin{pmatrix} M_{1,1}(X) & \cdots & M_{1,h}(X) \\ \vdots & \ddots & \vdots \\ M_{h,1}(X) & \cdots & M_{h,h}(X) \end{pmatrix}. \tag{3}$$

Under this formulation, each cross-head metric block $M_{i,j}(X)$ is no longer a matrix with constant parameter, but is dynamically generated conditioned on the input X. Further discussions on the advantages of our method are provided in Appendix A.

3.3 Attention Mechanism based on CHM

We propose DAMCHA, a novel attention mechanism built upon CHM. We assume that $M(X)$ has been generated by $f_\theta(\cdot)$ conditioned on the input sequence. DAMCHA consists of h attention heads. For a given input X, we first define the *row-summed head metric* for the i-th head as

$$\boldsymbol{m}_i(X) \;=\; \mathrm{RowSum}(M, i) \;=\; \sum_{j=1}^{h} M_{i,j}(X). \tag{4}$$

X is first linearly transformed to match the required dimensionality, Then, using $\boldsymbol{m}_i(X)$, DAMCHA computes the score matrix for the i-th head via a generalized quadratic form:

$$S_i(X) \;=\; \frac{1}{\sqrt{d_k}}\, X\, \boldsymbol{m}_i(X)\, X^\top \;\in\; \mathbb{R}^{L\times L}, \tag{5}$$

where $i = 1, 2, \ldots, h$, and d_k is set equal to the per-head dimension d_h, consistent with the scaling in conventional MHA. Then, the attention matrix is obtained by applying a softmax operation:

$$A_i(X) = \mathrm{Softmax}\left(S_i(X)\right) \in \mathbb{R}^{L\times L}. \tag{6}$$

For each head $i \in \{1, \ldots, h\}$, we compute value representations as $V_i = XW_V^{(i)}$ with projection $W_V^{(i)} \in \mathbb{R}^{d\times d_v}$. The output of the i-th head is given by

$$O_i(X) = A_i(X)\, V_i \in \mathbb{R}^{L\times d_v}, \tag{7}$$

where $A_i(X)$ denotes the attention weight and V_i is the corresponding value representation. The outputs from all heads are concatenated to form a joint output denoted as $O_{\mathrm{cat}}(X)$, given by

$$O_{\mathrm{cat}}(X) = \mathrm{Concat}\big(O_1(X), \ldots, O_h(X)\big). \tag{8}$$

Then, $O_{\mathrm{cat}}(X) \in \mathbb{R}^{L\times(hd_v)}$ is mapped back to the original feature dimension, yielding final output $X' = O_{\mathrm{cat}}(X)\, W_O \in \mathbb{R}^{L\times d}$ with projection $W_O \in \mathbb{R}^{(hd_v)\times d}$.

The RowSum operation preserves cross-head coupling information while offering two additional advantages. First, whereas each head metric $M_{i,i} = W_Q^{(i)}(W_K^{(i)})^\top$ in standard MHA is rank-limited by the projection dimension, summing h cross-head blocks elevates the effective rank upper bound, alleviating the low-rank bottleneck. Second, by compressing the full cross-head metric into a single effective matrix per head, RowSum preserves the standard multi-head output structure, allowing DAMCHA to serve as a natural drop-in replacement for MHA with negligible additional overhead. The workflows of DAMCHA and MHA are presented and compared in Figure 1. The data flow of DAMCHA is in Appendix B.

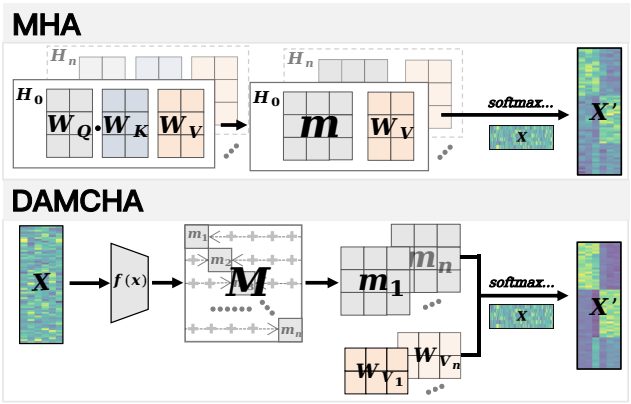

Figure 1: The workflows of DAMCHA and MHA.

3.4 Stack-Wise Sharing Mechanism

In practical applications, attention modules are typically embedded within network blocks that must be stacked to build depth. Learning separate $f_\theta(\cdot)$ mappings (each outputting a different metric matrix for a different block) is not only computationally expensive, but also prone to severe overfitting. To address this issue, we share the parameters of $f_\theta(\cdot)$ across all stacked network blocks. That is, all layers employ the same feedforward network to generate the adaptive CHMs. The stack-wise parameter sharing strategy of DAMCHA is further illustrated using a basic decoder-only Transformer architecture, as shown in Figure 2.

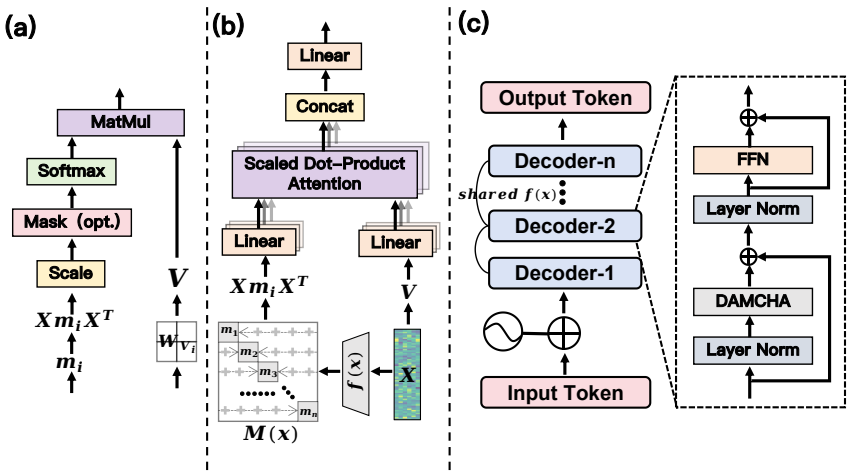

Figure 2: A basic DAMCHA-based decoder-only Transformer architecture with stack-wise sharing.

This stack-wise sharing yields multiple benefits. First, it substantially reduces the total number of parameters, especially when stacking many layers, thereby mitigating overfitting. Second, it introduces an additional form of regularization by constraining all layers to reuse a consistent family of metric transformations. This prevents excessive divergence of attention geometries across layers and avoids conflicts between layer-specific metric adaptations. Empirically, we observe that this shared design leads to more stable training and improved performance in deep architectures.

It is noted that, in DAMCHA, no explicit structural constraints or prior assumptions are imposed on any of $M_{i,j}(X)$. This is in sharp contrast to MHA, where the similarity matrices associated with individual heads, i.e., $M_{i,i}$, are inherently low-rank due to their dot-product formulation. By allowing all $M_{i,j}(X)$s to be freely learned without rank restrictions, DAMCHA effectively alleviates the low-rank bottleneck inherent in conventional attention mechanisms, further enhancing the expressive capacity of the model.

4 Theoretical View

We provide two theoretical results to justify DAMCHA, with proofs deferred to Appendix C. First, under mild expressivity assumptions on the metric generator, we prove that DAMCHA strictly generalizes standard MHA at the level of pre-softmax attention scores. This shows that DAMCHA is not merely a heuristic modification of MHA, but a strict functional extension. Second, we analyze DAMCHA from a metric-learning risk perspective. In heterogeneous data, different contexts may require different optimal similarity metrics. A static MHA metric must approximate all local geometries using a single global low-rank metric. In contrast, DAMCHA can condition its cross-head metric matrix on the input and select context-specific metrics. Under standard curvature and identifiability assumptions, the optimal risk of DAMCHA is no larger than that of MHA, with strict improvement whenever no single static low-rank metric can match all context-specific optimal metrics. For a detailed complexity analysis, see Appendix D.

5 Experiments

5.1 Decoder-Only Transformers with DAMCHA

Experimental Setting. We evaluate DAMCHA on both image and text generation tasks using decoder-only Transformers. For image generation, we train autoregressive models on CIFAR-10 and CIFAR-100, where images are flattened into sequences of $32 \times 32 = 1024$ tokens with 8-bit pixel values. For text generation, we use WMT14 [42] English-German translation and CommonGen [43] commonsense generation datasets. All models have 10 decoders, 8 attention heads, model dimension 512, and head dimension 64. Models are trained for 50 epochs with batch size 128, learning rate 3×10^{-4}, and Adam optimizer. We evaluate the performance using mean square error (MSE) and

Fréchet Inception Distance (FID) for images, and Negative Log-Likelihood (NLL) and BARTScore for text. To assess performance on high-quality samples, we additionally report metrics on the top-50 samples ranked by their generation likelihood. For DAMCHA, the metric generator f_θ is implemented as a 9-layer MLP, following the capacity ablation in Section 5.3.

Table 1 compares DAMCHA with MHA and several recent baselines, THA [25], DCMHA [19], ColMHA [26], MoA [27], and MMA [28]. DAMCHA achieves the best quality across both tasks.

Image Generation Performance. On CIFAR-10, DAMCHA obtains the lowest MSE mean (0.1210) and FID mean (331.8906), and substantially improves FID@Top-50 from 91.4320 under MHA to 63.0295, corresponding to a 31.1% relative reduction. Compared with the strongest non-DAMCHA baseline on FID@Top-50, i.e., MMA (84.9717), DAMCHA still achieves a 25.8% relative reduction. On CIFAR-100, DAMCHA also achieves the best FID mean (314.5722) and FID@Top-50 (53.7560), outperforming both MHA and recent attention variants.

Table 1: Comparison with MHA and recent multi-head attention variants on generation tasks. For MSE/FID/NLL, lower is better (↓). For BARTScore, higher is better (↑).

Image generation		MSE ↓			FID ↓			FID@Top-50 ↓
Dataset	Method	mean	q25	q75	mean	q25	q75	
CIFAR-10	MHA	0.1260	0.0750	0.1610	358.8570	293.0850	417.5810	91.4320
	THA	0.1357	0.0827	0.1726	366.7536	301.3989	421.4758	107.2156
	DCMHA	0.1264	0.0758	0.1622	366.0802	297.8571	418.4777	105.1269
	ColMHA	0.1276	0.0752	0.1630	365.9343	298.6780	416.1373	120.1717
	MMA	0.1292	0.0771	0.1604	359.6663	**267.3268**	422.7725	84.9717
	MoA	0.1576	0.0988	0.2036	368.4184	297.8040	416.2914	98.5777
	DAMCHA	**0.1210**	**0.0730**	**0.1547**	**331.8906**	297.1379	**411.8182**	**63.0295**
CIFAR-100	MHA	0.1250	0.0737	0.1600	358.1270	293.7460	418.5330	59.6860
	THA	0.1352	0.0823	0.1719	364.3847	302.4176	420.1534	57.9292
	DCMHA	0.1261	0.0754	0.1618	365.2419	298.4362	419.3781	79.9443
	ColMHA	0.1273	0.0748	0.1625	364.5126	297.2184	417.6439	58.1658
	MMA	0.1289	0.0767	0.1597	358.4291	291.4753	423.1847	55.6137
	MoA	0.1569	0.0981	0.2028	369.7253	298.3416	417.4832	88.2483
	DAMCHA	**0.1232**	**0.0730**	**0.1569**	**314.5722**	**288.9400**	**411.5314**	**53.7560**

Text generation		NLL ↓			BARTScore ↑			BARTScore@Top-50 ↑
Dataset	Method	mean	q25	q75	mean	q25	q75	
WMT	MHA	5.7730	5.1760	6.3750	-8.4740	-9.4590	-7.3420	-5.3090
	THA	5.2368	4.5104	5.7082	-6.7477	-7.5275	-5.8776	-4.5788
	DCMHA	5.2461	4.5396	5.7308	-6.8371	-7.5254	-6.0639	-4.8286
	ColMHA	5.2432	4.5387	5.7324	-6.1477	-6.7462	-5.4902	-4.3363
	MMA	**5.0301**	4.5096	5.6847	-8.3875	-9.3292	-7.3488	-5.7903
	MoA	5.2703	4.5695	5.7369	-6.9642	-7.6396	-6.2873	-5.0395
	DAMCHA	5.1907	**4.4945**	**5.6793**	**-5.4451**	**-6.0953**	**-5.1735**	**-4.1410**
CommonGen	MHA	5.7560	4.6510	6.7180	-5.9250	-6.4300	-5.2720	-4.2520
	THA	5.2287	4.3996	6.0161	-5.6318	-6.1313	**-4.9114**	-3.7457
	DCMHA	5.2754	4.3677	5.9989	-5.6686	**-5.7876**	-4.9677	-3.8115
	ColMHA	5.2210	4.3567	6.0294	-5.6745	-5.8728	-4.9503	-3.7936
	MMA	5.2952	**4.3092**	5.9861	-8.0765	-8.6621	-7.3931	-6.2185
	MoA	5.2918	4.2990	5.9834	-5.6805	-5.6899	-4.9597	-3.7899
	DAMCHA	**5.2204**	4.3105	**5.9788**	**-5.6142**	-6.1651	-4.9410	**-3.7417**

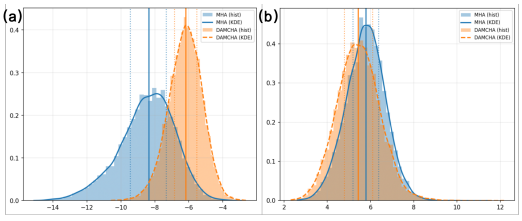

Figure 3: (a) Distribution of BARTScore on WMT data. (b) Distribution of NLL on WMT data.

Text Generation Performance. DAMCHA achieves the best BARTScore-based generation quality on both WMT and CommonGen. On WMT, DAMCHA improves BARTScore from -8.4740 under MHA to -5.4451 and improves BARTScore@Top-50 from -5.3090 to -4.1410. Compared with recent attention variants, DAMCHA also obtains the best BARTScore and BARTScore@Top-50, although MMA achieves a lower NLL mean. This suggests that lower token-level likelihood does not necessarily translate into better sequence-level generation quality. MMA enforces head diversity through fixed attention masks (global, local, forward, and backward patterns), which may reduce per-token prediction uncertainty and thus lower NLL, but its rigid structural constraints limit the model's ability to form flexible, context-sensitive attention patterns required for coherent long-range generation. BARTScore captures this distinction: MMA's forced diversity yields competitive NLL at the cost of substantially degraded holistic generation quality (BARTScore mean: -8.3875 for MMA versus -5.4451 for DAMCHA), precisely the dimension that DAMCHA's adaptive cross-head metric is designed to improve. On CommonGen, DAMCHA achieves the lowest NLL mean (5.2204), the best BARTScore mean (-5.6142), and the best BARTScore@Top-50 (-3.7417), further confirming the effectiveness of the proposed metric learning mechanism.

Performance in High Quality Generation A critical observation is that the improvement margins on Top-50 metrics are substantially larger than on average metrics. For instance, on WMT, BARTScore@Top-50 improves by 22.0% (-5.3090 to -4.1410) compared to 35.7% in average BARTScore. The improvements hold consistently across all tasks, suggesting that DAMCHA's input-adaptive metric learning particularly benefits the generation of high-quality, confident outputs where cross-head coordination is critical for capturing complex dependencies.

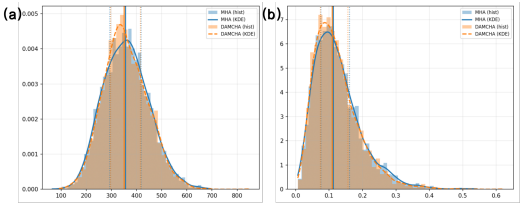

Figure 4: (a) Distribution of FID on CIFAR-10 data. (b) Distribution of MSE on data.CIFAR-10.

Table 2: Efficiency comparison with recent multi-head attention variants.

	CIFAR-10			CIFAR-100			CommonGen			WMT		
Method	Mem.	GFLOPs	Lat.(ms)	Mem.	GFLOPs	Lat.(ms)	Mem.	GFLOPs	Lat.(ms)	Mem.	GFLOPs	Lat.(ms)
THA	255.40	0.0797	**3.494**	255.40	0.0797	**3.535**	406.80	0.1845	3.611	1416.68	0.4698	7.299
DCMHA	292.71	0.0785	4.443	292.71	0.0785	4.499	463.36	0.1780	4.428	1650.30	0.4399	8.148
ColMHA	207.39	0.0765	3.576	207.39	0.0765	3.552	282.79	0.1699	**3.529**	903.74	0.4068	**3.868**
MMA	**191.39**	**0.0755**	16.677	**191.39**	**0.0755**	17.606	281.05	0.1678	25.863	923.74	0.4027	44.356
MoA	321.07	0.1101	4.078	321.07	0.1101	4.098	484.35	0.2538	4.095	1649.67	0.6417	6.770
DAMCHA	204.70	0.0786	3.868	196.57	0.0786	3.843	**271.23**	**0.1552**	3.844	**866.92**	**0.3586**	3.895

Figure 3 visualizes the distribution shifts on WMT samples. The BARTScore distribution (Figure 3(a)) shows a similar rightward shift with reduced variance. The NLL distribution (Figure 3(b)) of DAMCHA is notably tighter and shifted leftward compared to MHA, with the mode moving from approximately 5.8 to 5.2. Figure 4 visualizes the distribution shifts on the CIFAR-10 dataset and further shows that, under DAMCHA, the distributions of both MSE and FID are consistently more favorable than those obtained with MHA. These distributional changes indicate that DAMCHA not only improves average performance but also reduces uncertainty across the entire sample space, leading to more reliable generation quality. Additional distribution visualizations on CIFAR-100 and CommonGen are provided in Appendix E. We also visualize the evolution CHM during training in Appendix F. The results confirms that DAMCHA progressively learns to coordinate attention heads in input-dependent ways, moving beyond the block-diagonal constraint of standard MHA to a full Mahalanobis-like metric over the joint head space.

5.2 Efficiency and Computational Cost

We further evaluate DAMCHA and the baselines in terms of memory usage, GFLOPs, latency, and throughput. The results are reported in Table 2.

Image Generation Tasks. DAMCHA requires memory usage comparable to ColMHA and lower than THA, DCMHA, and MoA. For CIFAR-10, DAMCHA uses 0.0786 GFLOPs, which are very close to those of the best-performing baseline, MMA, while achieving substantially better FID and FID@Top-50. On CIFAR-100, DAMCHA similarly maintains low memory usage and competitive latency while delivering the best generation quality. Although THA and ColMHA achieve slightly higher throughput on some image settings, their generation metrics are consistently worse than DAMCHA, indicating that DAMCHA provides a better quality–efficiency balance.

Text Generation Tasks. The efficiency advantage becomes more pronounced on text generation tasks. On WMT, DAMCHA uses 866.92 memory units and 0.3586 GFLOPs, which are lower than THA, DCMHA, ColMHA, MMA, and MoA. It also achieves much lower latency than THA, DCMHA, MMA, and MoA. On CommonGen, DAMCHA also uses substantially less memory than THA, DCMHA, and MoA while maintaining competitive latency and throughput. These results show that DAMCHA's improvements are not simply obtained by increasing model size or computation, but are mainly attributable to the proposed cross-head metric learning design.

5.3 Ablation Study

Metric Learning & Stack-Wise Sharing. We first ablate the two key innovations of DAMCHA: (1) the input-adaptive metric generator $f(x)$ and (2) the stack-wise sharing mechanism. Table 4 presents results under three configurations: *w/o* $f(x)$*+share* removes both components, reducing DAMCHA to a static, unshared cross-head metric per-layer; *w/o share* retains input-adaptivity but uses layer-specific generators; and DAMCHA with both mechanisms.

The results demonstrate that both components are critical. Removing both $f(x)$ and sharing leads to the largest degradation—on CIFAR-10, FID@Top-50 increases by 17.59% (from 65.132 to 76.554), while average FID degrades by only 2.37%. This substantial gap confirms that input-adaptive metric learning is particularly crucial for high-quality generation. The *w/o share* variant shows moderate degradation, with FID@Top-50 increasing by 19.59% on CIFAR-10, indicating that independent per-layer causes overfitting and instability. Across all datasets, the pattern is consistent: removing input-adaptivity causes larger performance drops on Top-50 metrics than on average metrics. On

Table 3: Ablation on the capacity of the standard DAMCHA's $f(x)$ at different depth, and inference using full-, diagonal-only, and off-diagonal-only metric matrices. Within each row (fixed setting and dataset), the best layer is highlighted in red bold and the worst layer in blue for each metric block.

Setting	Dataset	FID@Top-50 / BART@Top-50					FID / BART				
		1-layer	3-layer	5-layer	7-layer	9-layer	1-layer	3-layer	5-layer	7-layer	9-layer
DAMCHA	CIFAR-10	77.97	75.79	71.07	74.07	**58.64**	250.35	231.52	234.37	**229.09**	229.84
	CIFAR-100	65.20	60.82	57.64	52.14	**50.29**	275.80	240.98	221.85	237.57	**217.89**
	WMT	-5.26	-5.14	-5.08	-4.98	**-4.51**	-7.71	-7.27	-6.00	-6.32	**-5.61**
	CommonGen	-4.97	-4.90	-4.62	-4.84	**-4.60**	-7.83	-6.53	-7.65	-6.90	**-6.20**
Diagonal-only	CIFAR-10	102.51	105.05	108.505	98.703	112.856	248.38	233.05	235.17	233.485	239.609
	CIFAR-100	85.763	69.73	76.102	59.685	61.945	275.535	227.707	236.587	231.397	237.241
	WMT	-6.218	-6.043	-5.622	-5.457	-5.618	-7.186	-7.411	-7.070	-7.854	-7.859
	CommonGen	-6.068	-5.213	-5.336	-4.865	-5.557	-7.674	-6.770	-7.612	-7.039	-6.956
Off-diagonal-only	CIFAR-10	97.318	140.37	87.217	97.647	126.921	245.87	252.75	250.201	249.907	241.249
	CIFAR-100	91.10	104.128	105.026	97.144	72.746	268.725	246.883	247.715	250.512	236.142
	WMT	-6.892	-6.101	-6.335	-5.997	-6.488	-8.095	-7.140	-8.112	-6.957	-8.119
	CommonGen	-5.10	-5.681	-5.954	-5.2457	-5.854	-6.926	-8.626	-8.913	-7.154	-8.467

CommonGen, BARTScore@Top-50 degrades by 13.99% without $f(x)$ compared to only 26.77% degradation in average BARTScore. Moreover, we provide a visualized gradient-flow analysis in Appendix G to illustrate how our stack-wise sharing improves learning performance.

Table 4: Ablation results. Here, $f(x)$ is implemented as an MLP. Blue ↓ indicates the relative degradation w.r.t. DAMCHA (in %).

Dataset	Variant	Metric	Metric@Top-50
CIFAR-10 (FID ↓)	w/o $f(x)$+share	233.038 ↓-2.37%	76.554 ↓-17.59%
	w/o share	228.961 ↓-0.58%	77.851 ↓-19.59%
	DAMCHA	**227.641**	**65.132**
CIFAR-100 (FID ↓)	w/o $f(x)$+share	229.467 ↓-8.66%	61.950 ↓-15.25%
	w/o share	234.296 ↓-10.95%	57.934 ↓-7.77%
	DAMCHA	**211.290**	**53.756**
WMT (BARTScore ↑)	w/o $f(x)$+share	-6.868 ↓-10.54%	-4.972 ↓-6.43%
	w/o share	-6.932 ↓-11.57%	-4.930 ↓-5.54%
	DAMCHA	**-6.213**	**-4.672**
CommonGen (BARTScore ↑)	w/o $f(x)$+share	-7.321 ↓-26.77%	-4.784 ↓-13.99%
	w/o share	-6.814 ↓-17.99%	-4.817 ↓-14.78%
	DAMCHA	**-5.775**	**-4.196**

Depth-wise Metric Learning & Cross-Head Learning. To analyze how metric learning strength affects learning performance and to verify the respective contributions of diagonal and off-diagonal blocks, we conduct three experiments: (1) standard DAMCHA using the full metric matrix, (2) diagonal-only inference where we zero out all off-diagonal blocks $M_{i,j}$ for $i \neq j$, and (3) off-diagonal-only inference where we zero out all diagonal blocks. For each setting, we evaluate the performance of $f(x)$ with different capacities (with 1, 3, 5, 7, 9 hidden layers).

Table 3 reveals two key findings. First, deeper layers consistently achieve better performance under the standard DAMCHA setting. For example, on CIFAR-10, FID@Top-50 improves from 77.97 at layer 1 to 58.64 at layer 9, while the average FID decreases from 250.35 to 229.84. This trend is consistent across all datasets, confirming that later layers learn increasingly task-relevant metric transformations. This hierarchical pattern aligns with the metric matrix evolution shown in Figure 8, where deeper layers develop more specialized interaction structures. These results indicate that appropriately designed metric learning is necessary for enhancing the expressive power of model.

Second, the diagonal-only inference results indicate that the block-diagonal constraint imposed by standard MHA is indeed limiting. Compared with full DAMCHA, diagonal-only inference exhibits a substantial performance degradation, particularly on the Top-50 metrics. On CIFAR-10, even the best diagonal-only configuration (layer 7, FID@Top-50 = 98.703) performs worse than the weakest full DAMCHA configuration (layer 1, FID@Top-50 = 77.97). Moreover, off-diagonal-only inference performs significantly worse than both full and diagonal-only metrics. This observation suggests that the data inherently contains interactions at cross-head level, and that cross-head interactions modeled through off-diagonal blocks are essential for capturing complex dependencies. More visualized results on metric geometry in ablations are presented in Appendix H.

5.4 Autoregressive Generation with GPT-2

We apply DAMCHA to the GPT-2 and conduct token-level pretraining experiments on three datasets: Enwik8 (100M characters of Wikipedia text available at `https://huggingface.co/google/reformer-enwik8`.), Text8 (100M characters of cleaned Wikipedia, available at `https://huggingface.co/datasets/afmck/text8`), and Mix (200M characters, a 50-50 mixture of

Enwik8 and Text8). All models have 12 layers, 12 attention heads, model dimension 768, and are trained for 100K steps with batch size 64, context length 512, and learning rate 6×10^{-4}. We report cross-entropy loss and perplexity (PPL) on held-out validation sets. Since DCMHA achieves the best performance among all baselines, we report only the comparison results against DCMHA and MHA.

Table 5 demonstrates that DAMCHA consistently outperforms both MHA and DCMHA across all three datasets. Notably, the gains are largest on the Mix dataset, where DAMCHA achieves 5.19% and 3.70% loss reduction over MHA and DCMHA respectively. This is consistent with DAMCHA's design rationale: the Mix dataset's heterogeneous semantic patterns demand cross-head coordination that MHA's block-diagonal structure cannot provide and DCMHA only partially addresses, while DAMCHA dynamically reconfigures its attention geometry to match the local context. Since DCMHA achieves the best performance among all baselines in the main generation experiments (Table 1), we focus our comparison on MHA and DCMHA to highlight the most informative contrast; the remaining baselines follow the same relative ordering observed in Section 5.1.

5.5 Discriminative Evaluation with Vision Transformers

We evaluate DAMCHA on ImageNet-1K classification using ViT-S/16 (12 layers, 6 heads, dimension 384, 22M parameters). Models are trained for 500 epochs with batch size 1024, learning rate 0.001, and standard augmentation. We report Top-1 and Top-5 accuracy at epochs 100 and 500 [44]. Table 6 shows that DAMCHA consistently outperforms both MHA and DCMHA baselines. DAMCHA achieves the highest accuracy across all evaluation points. At epoch 100, it obtains 70.93% Top-1 and 89.71% Top-5 accuracy, outperforming both MHA and DCMHA and indicating faster early-stage convergence. At epoch 500, DAMCHA further improves to 82.10% Top-1 and 96.23% Top-5 accuracy, maintaining consistent advantages over MHA and DCMHA. Although DCMHA narrows the gap after longer training, DAMCHA remains the best-performing model on all reported metrics. The larger early gap between DAMCHA and DCMHA suggests that DAMCHA's dynamic head coordination enables more efficient feature extraction from the start of training. While DCMHA narrows the gap with extended training, DAMCHA maintains advantages in both convergence speed and final performance across all metrics. Following the focused comparison strategy of Section 5.1, we report results against MHA and DCMHA, the strongest baseline across the generation experiments.

Table 5: Token-level pretraining results. Lower is better for cross-entropy loss and PPL. Red ↑ denotes DAMCHA's relative improvement over DCMHA (in %).

Dataset	Cross-entropy loss ↓			PPL ↓		
	MHA	DCMHA	DAMCHA	MHA	DCMHA	DAMCHA
Enwik8	1.07	1.05	**1.03** ↑1.90%	2.90	2.86	**2.80** ↑2.10%
Text8	1.06	1.05	**1.03** ↑1.90%	2.88	2.86	**2.81** ↑1.75%
Mix	1.09	1.08	**1.04** ↑3.70%	2.98	2.94	**2.82** ↑4.08%

5.6 Experiments with Synthetic Tasks

Table 6: Results on ImageNet-1K classification.

Model	Top-1		Top-5	
	epoch 100	epoch 500	epoch 100	epoch 500
MHA	69.02%	79.51%	88.68%	94.68%
DCMHA	67.90%	81.70%	88.71%	95.00%
DAMCHA	**70.93%**	**82.10%**	**89.71%**	**96.23%**

To further understand the mechanism underlying DAMCHA's improvements, we design two synthetic tasks that explicitly require coordinated attention across multiple subspaces. These controlled experiments reveal that DAMCHA's advantage grows with the strength of cross-subspace entanglement: DAMCHA's relative gain increases and scales with feature dimensionality in the classification task. Crucially, when cross-subspace interactions are weak or absent, DAMCHA remains competitive with MHA, confirming that the adaptive cross-head metric incurs no penalty in simpler settings. These results provide controlled evidence that DAMCHA's improvements on real-world benchmarks are driven by its ability to model dependencies that span multiple attention heads, rather than by an increase in parameter count or computational cost. Full experimental details and results are provided in Appendix I. Computational resource details are reported in Appendix J, and a mechanistic discussion of DAMCHA's advantages over competing baselines is provided in Appendix K.

6 Conclusion

This paper proposes a novel attention mechanism, DAMCHA. By lifting both the block-diagonal and input-independent constraints of MHA, DAMCHA generalizes it into a data-adaptive metric learning framework. This formulation enables dynamic reconfiguration of attention geometry and explicit cross-head interaction modeling, providing significantly enhanced representational and approximation capabilities. Extensive experiments consistently demonstrate that DAMCHA delivers stronger performance, with particularly pronounced gains on high-quality and high-confidence samples. These results confirm that learning input-adaptive cross-head metrics for attention effectively enhances the representational capacity of Transformers, making DAMCHA well suited for complex dependency modeling and generative tasks.

Acknowledgments

This work was supported by Innovation and Talent Base for Digital Technology and Finance (B21038), Fundamental Research Funds for the Central Universities, China under Project (Zhongnan University of Economics and Law, China) 2722024EJ011 and 2722022BY020, and General Research Fund grants 14303724 from Research Grant Council of Hong Kong

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010.

[2] P. Michel, O. Levy, G. Neubig, Are sixteen heads really better than one?, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, pp. 14014–14024.

[3] L. Liu, J. Liu, J. Han, Multi-head or single-head? an empirical comparison for transformer training, arXiv preprint arXiv:2106.09650ACL ARR 2022 February Blind Submission (2021).

[4] T. Ruan, S. Zhang, Towards understanding how attention mechanism works in deep learning, arXiv preprint arXiv:2412.18288 (2024).

[5] Y. Shen, E. M.-K. Lai, M. Mohaghegh, Effects of similarity score functions in attention mechanisms on the performance of neural question answering systems, Neural Processing Letters 54 (3) (2022) 2283–2302.

[6] J.-H. Kim, J. Jun, B.-T. Zhang, Bilinear attention networks, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, 2018, pp. 1571–1581.

[7] Y. Bian, J. Huang, X. Cai, J. Yuan, K. Church, On attention redundancy: A comprehensive study, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 930–945.

[8] S. Liu, H. Peng, Z. Zhang, Z. Chen, Y. Guo, Multi-head low-rank attention, in: NeurIPS 2025 Workshop on Efficient Reasoning, 2025.

[9] C. Shen, J. Kim, L. Wang, Scalable large-margin mahalanobis distance metric learning, IEEE Transactions on Neural Networks 21 (9) (2010) 1524–1530.

[10] J. Liu, Y. Qiao, Mahalanobis distance–based kernel supervised machine learning in spectral dimensionality reduction for hyperspectral imaging remote sensing, International Journal of Distributed Sensor Networks 16 (11) (2020).

[11] Y. Zhao, C. Thrampoulidis, Why loss re-weighting works if you stop early: Training dynamics of unconstrained features, in: ICML 2025 Workshop on Methods and Opportunities at Small Scale, 2025.

[12] S. Zhong, W. Ruan, M. Jin, H. Li, Q. Wen, Y. Liang, Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting, in: Forty-second International Conference on Machine Learning, 2025.

[13] H. Zhang, Q. Jin, T. Hartvigsen, M. Udler, M. Ghassemi, A pipeline for interpretable clinical subtyping with deep metric learning, in: ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH), 2023.

[14] A. Galassi, M. Lippi, P. Torroni, Attention in natural language processing, IEEE Transactions on Neural Networks and Learning Systems 32 (10) (2021) 4291–4308.

[15] S. Chaudhari, V. Mithal, G. Polatkan, R. Ramanath, An attentive survey of attention models, ACM Transactions on Intelligent Systems and Technologies 12 (5) (2021) 1–32.

[16] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, M. Shah, Transformers in vision: A survey, ACM Computing Surveys 54 (10s) (2022) 1–41.

[17] J. Merullo, C. Eickhoff, E. Pavlick, Talking heads: Understanding inter-layer communication in transformer language models, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[18] J.-B. Cordonnier, A. Loukas, M. Jaggi, Multi-head attention: Collaborate instead of concatenate (2021).

[19] D. Xiao, Q. Meng, S. Li, X. Yuan, Improving transformers with dynamically composable multi-head attention, in: Proceedings of the 41st International Conference on Machine Learning, Vol. 235 of Proceedings of Machine Learning Research, PMLR, 2024.

[20] Z. Zhang, N. Shao, C. Gao, R. Miao, Q. Yang, J. Shao, Mixhead: Breaking the low-rank bottleneck in multi-head attention language models, Knowledge-Based Systems 240 (C) (2022) 108075.

[21] Q. Guo, X. Qiu, X. Xue, Z. Zhang, Low-rank and locality constrained self-attention for sequence modeling, IEEE/ACM Transactions on Audio, Speech, and Language Processing 27 (12) (2019) 2213–2222.

[22] S. Bhojanapalli, C. Yun, A. S. Rawat, S. J. Reddi, S. Kumar, Low-rank bottleneck in multi-head attention models, arXiv preprint arXiv:2002.07028 (2020).

[23] V. N. Karkaria, D. Lee, Y.-P. Chen, Y. Yu, W. Chen, ASNO: An interpretable attention-based spatio-temporal neural operator for robust scientific machine learning, in: ICML 2025 Workshop on Reliable and Responsible Foundation Models, 2025.

[24] M. Yang, A. Feng, B. Xiong, J. Liu, I. King, R. Ying, Enhancing LLM complex reasoning capability through hyperbolic geometry, in: ICML 2024 Workshop on LLMs and Cognition, 2024.

[25] N. Shazeer, Z. Lan, Y. Cheng, N. Ding, L. Hou, Talking-heads attention, arXiv preprint arXiv:2003.02436 (2020).

[26] J.-B. Cordonnier, A. Loukas, M. Jaggi, Multi-head attention: Collaborate instead of concatenate, arXiv preprint arXiv:2006.16362 (2020).

[27] X. Zhang, Y. Shen, Z. Huang, J. Zhou, W. Rong, Z. Xiong, Mixture of attention heads: Selecting attention heads per token, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2022, pp. 4150–4162.

[28] H. Cui, S. Iida, P.-H. Hung, T. Utsuro, M. Nagata, Mixed multi-head self-attention for neural machine translation, in: Proceedings of the 3rd Workshop on Neural Generation and Translation, Association for Computational Linguistics, 2019, pp. 206–214.

[29] Y. Li, J. Wang, X. Dai, L. Wang, C.-C. M. Yeh, Y. Zheng, W. Zhang, K.-L. Ma, How does attention work in vision transformers? a visual analytics attempt, arXiv preprint arXiv:2303.13731 (2023).

[30] S. Jo, G. Jang, H. Park, Gmar: Gradient-driven multi-head attention rollout for vision transformer interpretability, in: Proceedings of the 2025 IEEE International Conference on Image Processing, 2025, pp. 582–587.

[31] G. Han, S. Shen, Research on chatglm-based tower machine intelligent q&a system with cross-layer shared dynamic attention, in: Proceedings of the 2025 7th International Conference on Natural Language Processing, 2025, pp. 200–204.

[32] Y.-H. Lan, Gateformer: Advancing multivariate time series forecasting via temporal and variate-wise attention with gated representations, in: 1st ICML Workshop on Foundation Models for Structured Data, 2025.

[33] X. Wang, Y. Hua, E. Kodirov, G. Hu, N. M. Robertson, Deep metric learning by online soft mining and class-aware attention, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 6129–6136.

[34] X. Liu, Z. Guo, J. You, B. V. K. V. Kumar, Dependency-aware attention control for image set-based face recognition, IEEE Transactions on Information Forensics and Security 15 (2020) 1501–1512.

[35] Y. Tay, L. A. Tuan, S. C. Hui, Latent relational metric learning via memory-based attention for collaborative ranking, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 729–739.

[36] Z. Yang, K. Wang, L. Ye, X. Yuan, Y. Wang, C. Yang, W. Gui, A difference metric attention with position distance-based weighting for transformer in data sequence modeling of industrial processes, IEEE Transactions on Industrial Informatics 21 (2) (2025) 1803–1812.

[37] Y. Huang, Z. Wen, Y. Chi, Y. Liang, How transformers learn diverse attention correlations in masked vision pretraining, in: ICML 2024 Workshop on Theoretical Foundations of Foundation Models, 2024.

[38] S. Wang, K. Wang, W. Wang, Rauf: Learning the spatial uncertainty field of radar, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026, pp. 42474–42483.

[39] B. Koyuncu, T. N. Bauerschmidt, I. Valera, E-protran: Efficient probabilistic transformers for forecasting, in: ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling, 2024.

[40] T. Y. Chen, Dynamic causal-graph memory: Structured retrieval for million–token reasoning, in: ICML 2025 Workshop on Long-Context Foundation Models, 2025.

[41] X. Liu, Z. Guo, J. You, B. V. K. V. Kumar, Attention control with metric learning alignment for image set-based recognition (2019).

[42] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, A. Tamchyna, Findings of the 2014 workshop on statistical machine translation, in: Proceedings of the Ninth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Baltimore, Maryland, USA, 2014, pp. 12–58.

[43] B. Y. Lin, W. Zhou, M. Shen, P. Zhou, C. Bhagavatula, Y. Choi, X. Ren, Commongen: A constrained text generation challenge for generative commonsense reasoning, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, 2020, pp. 1823–1840.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.

Appendix

A Further Illustration on Expressive Power and Advantages of CHM

The introduction of cross-head interactions significantly enhances representational flexibility. For example, when the similarity between two tokens depends on interactions between features residing in different subspaces, standard MHA may fail to capture such dependencies if those features are assigned to separate heads. The CHM directly models these interactions via the off-diagonal blocks $M_{i,j}$, enabling explicit cross-subspace similarity estimation. From a theoretical standpoint, removing the block-diagonal constraint allows the attention score matrix to represent higher-rank correlation patterns. As a result, a single attention layer equipped with cross-head metrics can express information structures that would otherwise require multiple stacked layers in standard MHA, leading to more efficient and powerful representation learning.

For our adaptive learning of CHM, the formulation can be interpreted as introducing a form of conditional metric learning, where the attention geometry is modulated by the current context. Rather than learning a single global notion of similarity, the model learns how to select or generate an appropriate metric given the input. Functionally, this resembles dynamic parameterization or conditional computation, in which a lightweight network produces part of the parameters of the main model to enable context-dependent behavior. From a representational standpoint, generating $M(X)$ via $f(\cdot)$ explicitly introduces nonlinear, higher-order interactions into attention. The resulting attention score depends not only on the quadratic form $X^\top MX$, but also on the nonlinear transformation induced by f. As such, the attention mechanism can be viewed as operating on higher-order polynomial features of X, substantially enriching its function approximation and expressive capacity.

Several advantages are expected to be provided by our method Context-sensitive attention: The model can dynamically adjust its attention patterns across different inputs, improving sensitivity to contextual variations. Enhanced expressiveness: By allowing M to depend on X, the attention mechanism is no longer restricted to a fixed bilinear form, but instead realizes a composite mapping, which spans a significantly richer function class and can capture more complex dependencies. Reduced approximation bias: The adaptive mechanism alleviates the bias introduced by static parameters. Even if certain relationships are not learned as dominant global patterns during training, the model can still approximate them for specific inputs by generating specialized attention metrics, thereby reducing representational blind spots.

Asymmetric Design of Value Projection. A natural question concerns the asymmetric treatment of the query–key pathway versus the value pathway in DAMCHA: while f_θ replaces conventional linear query–key projections with an input-adaptive metric generator, the value projection $W_V^{(i)}$ retains the standard static linear form of MHA. This design is deliberate. The query–key pathway determines *where* attention is directed, that is, the similarity geometry, which varies most across contexts and benefits most from input adaptivity. The value projection, by contrast, determines *what* information is aggregated given fixed attention weights, a role that is well served by a static linear transformation in practice. Extending adaptivity to the value pathway would add computational overhead and optimization complexity without the same theoretical motivation: the expressiveness gains in Theorems C.1 and C.2 arise specifically from the adaptive similarity structure in the pre-softmax score, not from adaptive value representations. Exploring adaptive value projections in combination with DAMCHA's metric learning is nonetheless an interesting direction for future work.

B Data Flow and Tensor Shapes of DAMCHA

Figure 5 summarizes the end-to-end data flow of DAMCHA and MHA.

C Theoretical Analysis

In this section, we provide a theoretical justification for DAMCHA from two complementary perspectives. First, we show that the score-function class induced by DAMCHA strictly contains that of standard MHA. Second, we interpret both MHA and DAMCHA through the lens of metric learning

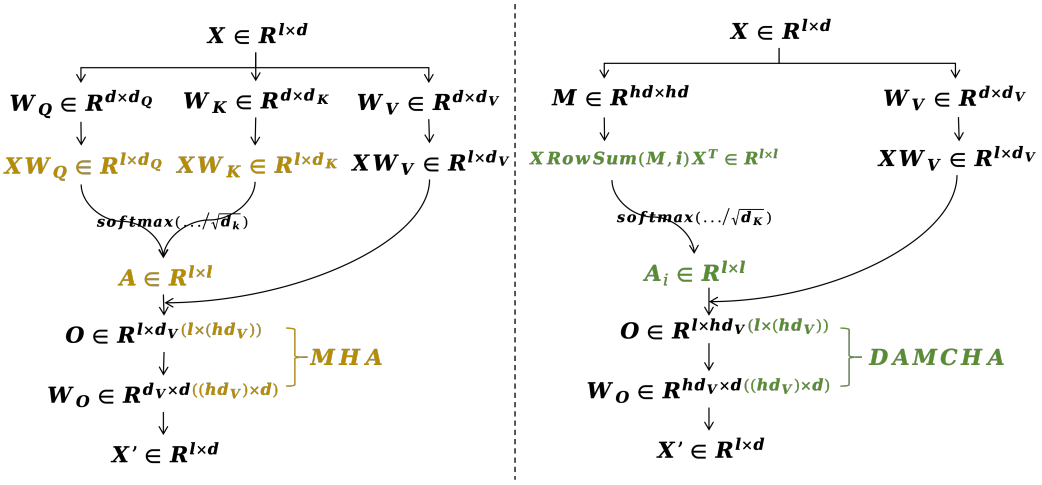

Figure 5: Data flow and tensor shapes of DAMCHA (right, with the value/output projection shapes consistent with standard MHA)

and prove that input-adaptive cross-head metrics can reduce the irreducible metric mismatch incurred by a static global metric.

C.1 DAMCHA Strictly Generalizes MHA

We first formalize the attention score functions induced by standard MHA and DAMCHA. Let $X \in \mathbb{R}^{L\times d}$ denote an input sequence, where L is the sequence length and $d = d_{\text{model}}$. For the i-th head of standard MHA, the pre-softmax score matrix can be written as

$$S_i^{\text{MHA}}(X) = \frac{1}{\sqrt{d_h}} X M_i X^\top, \qquad M_i = W_Q^{(i)} (W_K^{(i)})^\top, \tag{9}$$

where $W_Q^{(i)}, W_K^{(i)} \in \mathbb{R}^{d\times d_h}$. Thus, each head in MHA corresponds to a static low-rank bilinear similarity metric and $\text{rank}(M_i) \le d_h$. For DAMCHA, the i-th head is given by

$$\bar{M}_i(X) = \sum_{j=1}^{h} M_{i,j}(X), \tag{10}$$

where $M_{i,j}(X)$ is generated by the input-dependent metric generator $f_\theta(\cdot)$. The corresponding pre-softmax score matrix is

$$S_i^{\text{DAMCHA}}(X) = \frac{1}{\sqrt{d_k}} X \bar{M}_i(X) X^\top. \tag{11}$$

We define the score-function class induced by standard MHA as

$$\mathcal{S}_{\text{MHA}} = \left\{ X \mapsto \left(XM_1X^\top, \ldots, XM_hX^\top \right) : M_i = A_i B_i^\top,\ A_i, B_i \in \mathbb{R}^{d\times d_h} \right\}. \tag{12}$$

Similarly, the score-function class induced by DAMCHA is defined as

$$\mathcal{S}_{\text{DAMCHA}} = \left\{ X \mapsto \left(X\bar{M}_1(X)X^\top, \ldots, X\bar{M}_h(X)X^\top \right) : \bar{M}_i(X) = \sum_{j=1}^{h} M_{i,j}(X) \right\}. \tag{13}$$

We impose the following mild expressivity assumption on the metric generator.

Assumption 1. The metric generator f_θ is capable of representing constant matrix mappings. That is, for any collection of matrices $\{B_{i,j}\}_{i,j=1}^h$, there exists a parameter θ such that $M_{i,j}(X) \equiv B_{i,j}$, for all X. This assumption is mild because a standard feedforward network or hypernetwork with bias terms can represent constant functions.

Theorem C.1 (Strict Function-Class Generalization of MHA)**.** *Suppose Assumption 1 holds and $d_h < d$. Then the score-function class induced by standard MHA is strictly contained in that induced by DAMCHA:*

$$\mathcal{S}_{\mathrm{MHA}} \subsetneq \mathcal{S}_{\mathrm{DAMCHA}}.$$

Consequently, DAMCHA strictly generalizes MHA at the level of pre-softmax attention score functions.

Proof. We prove the result in two steps. **Step 1: Inclusion.** We first show that $\mathcal{S}_{\mathrm{MHA}} \subseteq \mathcal{S}_{\mathrm{DAMCHA}}$. By Assumption 1, DAMCHA can choose its generated cross-head metric blocks as constant matrices satisfying $M_{i,i}(X) \equiv M_i$ and $M_{i,j}(X) \equiv 0$, $j \neq i$. Then the row-summed effective metric of the i-th head becomes

$$\bar{M}_i(X) = \sum_{j=1}^{h} M_{i,j}(X) = M_{i,i}(X) = M_i. \tag{14}$$

It follows that

$$X\bar{M}_i(X)X^\top = XM_iX^\top. \tag{15}$$

Then, every score function produced by MHA can be exactly reproduced by DAMCHA. This proves

$$\mathcal{S}_{\mathrm{MHA}} \subseteq \mathcal{S}_{\mathrm{DAMCHA}}. \tag{16}$$

Step 2: Strictness. We now show that the inclusion is strict. Since $d_h < d$, there exists a matrix $B \in \mathbb{R}^{d\times d}$ such that $\mathrm{rank}(B) > d_h$. By Assumption 1, DAMCHA can choose a constant effective metric for the first head as

$$\bar{M}_1(X) \equiv B. \tag{17}$$

For instance, this can be achieved by setting $M_{1,1}(X) \equiv B$, $M_{1,j}(X) \equiv 0$, $j \neq 1$. Then DAMCHA induces the score function $X \mapsto XBX^\top$. Suppose, for contradiction, that this score function can also be represented by standard MHA. Then there must exist matrices $A, B' \in \mathbb{R}^{d\times d_h}$ such that $M = A(B')^\top$ and $XMX^\top = XBX^\top$ for all admissible inputs X. Equivalently, for all token vectors $u, v \in \mathbb{R}^d$, we have $u^\top Mv = u^\top Bv$, which implies $M = B$. However, since $M = A(B')^\top$, we have

$$\mathrm{rank}(M) \leq d_h. \tag{18}$$

This contradicts

$$\mathrm{rank}(B) > d_h. \tag{19}$$

Then, the score function $X \mapsto XBX^\top$ belongs to $\mathcal{S}_{\mathrm{DAMCHA}}$ but not to $\mathcal{S}_{\mathrm{MHA}}$. Hence,

$$\mathcal{S}_{\mathrm{MHA}} \subsetneq \mathcal{S}_{\mathrm{DAMCHA}}. \tag{20}$$

The proof is complete. □

C.2 Metric-Learning Advantage of Dynamic Cross-Head Metrics

The previous result establishes that DAMCHA strictly enlarges the attention score-function class. We next explain why this enlargement is meaningful from the perspective of metric learning.

The key difference between MHA and DAMCHA is that MHA learns a static global metric, whereas DAMCHA learns an input-adaptive metric. In heterogeneous data, the optimal similarity geometry may vary across contexts. A single global metric must average over these different local geometries and therefore suffers from an irreducible metric mismatch. DAMCHA can reduce this mismatch by conditioning the metric on the input.

Let $C \in \{1, \ldots, K\}$ denote a latent context variable with

$$\mathbb{P}(C = c) = \pi_c > 0, \qquad \sum_{c=1}^{K} \pi_c = 1. \tag{21}$$

The context C may represent different semantic regimes, syntactic patterns, visual structures, or distributional modes. We assume that, for each context c, there exists a context-specific optimal collection of effective attention metrics

$$\mathbf{M}_c^\star = \left(M_{1,c}^\star, \ldots, M_{h,c}^\star\right), \tag{22}$$

where $M_{i,c}^\star \in \mathbb{R}^{d\times d}$ denotes the optimal effective metric for the i-th head under context c. Let $\ell(\mathbf{M}; X, Y)$ denote a generic metric-learning loss associated with the effective metrics

$$\mathbf{M} = (M_1, \ldots, M_h). \tag{23}$$

This loss may correspond to attention alignment loss, contrastive metric loss, classification loss, or a downstream task loss induced by the attention scores. For each context c, define the conditional risk

$$r_c(\mathbf{M}) = \mathbb{E}\left[\ell(\mathbf{M}; X, Y) \mid C = c\right]. \tag{24}$$

We impose the following assumptions.

Assumption 2. For each context c, the conditional risk $r_c(\mathbf{M})$ has a unique minimizer $\mathbf{M}_c^\star$, namely

$$\mathbf{M}_c^\star = \arg\min_{\mathbf{M}} r_c(\mathbf{M}). \tag{25}$$

Assumption 3. There exists a constant $\mu > 0$ such that, for every context c and every collection of metrics $\mathbf{M} = (M_1, \ldots, M_h)$,

$$r_c(\mathbf{M}) - r_c(\mathbf{M}_c^\star) \geq \frac{\mu}{2} \sum_{i=1}^{h} \left\|M_i - M_{i,c}^\star\right\|_F^2. \tag{26}$$

This condition states that the conditional risk increases at least quadratically as the learned metric deviates from the context-specific optimal metric.

Assumption 4. The context C is identifiable from the input X, namely there exists a measurable function $q(\cdot)$ such that

$$C = q(X). \tag{27}$$

Moreover, the DAMCHA metric generator is expressive enough to represent the context-dependent mapping

$$X \mapsto \mathbf{M}_{q(X)}^\star. \tag{28}$$

Assumption 4 formalizes the idea that the input contains sufficient information to determine the appropriate local similarity geometry. For standard MHA, the effective metric of each head is static and belongs to the low-rank query-key factorization class

$$\mathcal{L}_{d_h} = \left\{AB^\top : A, B \in \mathbb{R}^{d\times d_h}\right\}. \tag{29}$$

Thus, the optimal static MHA risk is

$$R_{\mathrm{MHA}}^\star = \inf_{M_i \in \mathcal{L}_{d_h}} \sum_{c=1}^{K} \pi_c r_c(M_1, \ldots, M_h). \tag{30}$$

In contrast, DAMCHA can use input-adaptive effective metrics

$$\mathbf{M}(X) = \left(\bar{M}_1(X), \ldots, \bar{M}_h(X)\right), \tag{31}$$

where each $\bar{M}_i(X)$ is generated through the cross-head metric matrix. The optimal DAMCHA risk is defined as

$$R_{\mathrm{DAMCHA}}^\star = \inf_{g} \mathbb{E}\left[\ell(g(X); X, Y)\right], \tag{32}$$

where $g(X)$ ranges over the metric mappings representable by DAMCHA.

Theorem C.2 (Metric-Learning Risk Advantage of DAMCHA)**.** *Suppose Assumptions 2–4 hold. Then*

$$R^\star_{\mathrm{DAMCHA}} \le R^\star_{\mathrm{MHA}}. \tag{33}$$

Moreover,

$$R^\star_{\mathrm{MHA}} - R^\star_{\mathrm{DAMCHA}} \ge \frac{\mu}{2} \inf_{M_i \in \mathcal{L}_{d_h}} \sum_{c=1}^{K} \pi_c \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 . \tag{34}$$

Consequently, if there does not exist a single static collection of low-rank metrics $(M_1, \dots, M_h)$, with $M_i \in \mathcal{L}_{d_h}$, such that

$$M_i = M^\star_{i,c} \qquad \textit{for all } i = 1, \dots, h \textit{ and all } c = 1, \dots, K, \tag{35}$$

then

$$R^\star_{\mathrm{DAMCHA}} < R^\star_{\mathrm{MHA}}. \tag{36}$$

Proof. We first characterize the best risk attainable by an input-adaptive metric. By Assumption 4, DAMCHA can represent the mapping $g^\star(X) = \mathbf{M}^\star_{q(X)}$. Since $C = q(X)$, this means that DAMCHA can select $g^\star(X) = \mathbf{M}^\star_C$ for each input. Therefore, the risk achieved by this oracle adaptive metric is

$$\mathbb{E}\left[\ell(g^\star(X); X, Y)\right] = \sum_{c=1}^{K} \pi_c \mathbb{E}\left[\ell(\mathbf{M}^\star_c; X, Y) \mid C = c\right]. \tag{37}$$

By the definition of $r_c(\cdot)$, this becomes

$$\mathbb{E}\left[\ell(g^\star(X); X, Y)\right] = \sum_{c=1}^{K} \pi_c r_c(\mathbf{M}^\star_c). \tag{38}$$

Since $R^\star_{\mathrm{DAMCHA}}$ is the infimum over all DAMCHA-representable mappings, and $g^\star$ is one such mapping, we have

$$R^\star_{\mathrm{DAMCHA}} \;\le\; \mathbb{E}\big[\ell(g^\star(X);\, X, Y)\big] = \sum_{c=1}^{K} \pi_c\, r_c(\mathbf{M}^\star_c). \tag{39}$$

Now consider standard MHA. It must use a static collection of metrics

$$\mathbf{M} = (M_1, \dots, M_h), \qquad M_i \in \mathcal{L}_{d_h}, \tag{40}$$

independent of the input context. For any such $\mathbf{M}$, its excess risk over the adaptive oracle is

$$\sum_{c=1}^{K} \pi_c r_c(\mathbf{M}) - \sum_{c=1}^{K} \pi_c r_c(\mathbf{M}^\star_c) = \sum_{c=1}^{K} \pi_c \left[r_c(\mathbf{M}) - r_c(\mathbf{M}^\star_c) \right]. \tag{41}$$

By Assumption 3,

$$r_c(\mathbf{M}) - r_c(\mathbf{M}^\star_c) \ge \frac{\mu}{2} \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 . \tag{42}$$

Therefore,

$$\sum_{c=1}^{K} \pi_c r_c(\mathbf{M}) - \sum_{c=1}^{K} \pi_c r_c(\mathbf{M}^\star_c) \ge \frac{\mu}{2} \sum_{c=1}^{K} \pi_c \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 . \tag{43}$$

Taking the infimum over all static MHA metrics $M_i \in \mathcal{L}_{d_h}$, we obtain

$$R^\star_{\mathrm{MHA}} - R^\star_{\mathrm{DAMCHA}} \ge \frac{\mu}{2} \inf_{M_i \in \mathcal{L}_{d_h}} \sum_{c=1}^{K} \pi_c \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 . \tag{44}$$

The right-hand side is nonnegative, which proves $R^\star_{\mathrm{DAMCHA}} \le R^\star_{\mathrm{MHA}}$. Furthermore, if no single static low-rank collection $(M_1, \dots, M_h)$ can simultaneously match all context-specific optimal metrics $\mathbf{M}^\star_c$, then

$$\inf_{M_i \in \mathcal{L}_{d_h}} \sum_{c=1}^{K} \pi_c \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 > 0. \tag{45}$$

Therefore, we have $R^\star_{\mathrm{MHA}} - R^\star_{\mathrm{DAMCHA}} > 0$, which implies $R^\star_{\mathrm{DAMCHA}} < R^\star_{\mathrm{MHA}}$. □

Theorem C.2 provides a metric-learning explanation for the advantage of DAMCHA. Standard MHA learns a single static metric shared by all inputs. When the data distribution contains multiple contexts with different optimal similarity geometries, this static metric must compromise among them and therefore suffers from an irreducible metric mismatch:

$$\inf_{M_i \in \mathcal{L}_{d_h}} \sum_{c=1}^{K} \pi_c \sum_{i=1}^{h} \left\| M_i - M^\star_{i,c} \right\|_F^2 . \tag{46}$$

In contrast, DAMCHA can condition its cross-head metric matrix on the input and select a context-specific effective metric. Thus, DAMCHA behaves as an input-adaptive local metric learner, whereas MHA behaves as a static global metric learner.

D Computational Complexity Analysis of DAMCHA

D.1 Notation

Throughout this section we use the following symbols consistently.

Symbol	Meaning
L	Sequence (context) length
d	Model dimension d_{model}
h	Number of attention heads
d_h	Per-head dimension, $d_h = d/h$
d_v	Per-head value dimension (set equal to d_h unless noted)
P	Depth (number of hidden layers) of the metric generator f_θ
w	Hidden width of f_θ
N	Number of stacked Transformer layers (depth of the network)

All asymptotic statements use the standard $\mathcal{O}(\cdot)$ notation and suppress constant factors. We analyse one DAMCHA attention layer and one standard MHA layer side-by-side, then aggregate to the full N-layer stack.

D.2 Per-Layer Time Complexity

A single DAMCHA attention layer consists of five distinct computational stages.

Stage 1: Metric generator $f_\theta(X)$. Given the input sequence $X \in \mathbb{R}^{L \times d}$, the metric generator first computes a context summary $\bar{x} = \frac{1}{L}\sum_{\ell=1}^{L} X_\ell \in \mathbb{R}^d$ via mean pooling, at cost $\mathcal{O}(Ld)$. The resulting vector is passed through a P-layer MLP with hidden width w:

$$f_\theta : \mathbb{R}^d \xrightarrow{d \to w} \left(\mathbb{R}^w\right)^{P-2} \xrightarrow{w \to d^2} \mathbb{R}^{d^2},$$

producing the flattened cross-head metric matrix $M(X) \in \mathbb{R}^{d \times d}$. The dominant costs are:

$$\begin{aligned} \text{Input projection :} \quad & \mathcal{O}(d \cdot w), \\ (P-2)\ \text{hidden layers :} \quad & \mathcal{O}((P-2)\, w^2), \\ \text{Output projection :} \quad & \mathcal{O}(w \cdot d^2). \end{aligned}$$

Summing all stages of f_θ (including the pooling step):

$$T_{f_\theta} = \mathcal{O}\big(Ld + Pw^2 + wd^2\big)\,. \tag{47}$$

Crucially, under stack-wise parameter sharing f_θ is shared across all N layers and need only be evaluated *once* per forward pass, so its amortised per-layer cost is T_{f_θ}/N.

Stage 2: RowSum aggregation. The cross-head metric $M(X)$ is an $h \times h$ block matrix with blocks $M_{i,j}(X) \in \mathbb{R}^{d_h \times d_h}$. For each query head i we compute

$$m_i(X) = \sum_{j=1}^{h} M_{i,j}(X) \ \in \ \mathbb{R}^{d_h \times d_h}.$$

This requires summing h matrices of size $d_h \times d_h$ for each of h heads:

$$T_{\text{RowSum}} = \mathcal{O}\big(h^2 d_h^2\big) = \mathcal{O}\big(d^2\big)\,, \tag{48}$$

since $h \cdot d_h = d$.

Stage 3: Attention score computation. For each head i, let $X_i = X[:, id_h : (i{+}1)d_h] \in \mathbb{R}^{L \times d_h}$ be the i-th positional slice of the input. The pre-softmax score matrix is

$$S_i(X) = \frac{1}{\sqrt{d_k}}\, X_i\, m_i(X)\, X_i^\top \ \in \ \mathbb{R}^{L \times L}.$$

The two matrix products cost:

$$\begin{aligned} X_i m_i(X): &\quad \mathcal{O}(L d_h^2), \\ \big(X_i m_i(X)\big) X_i^\top: &\quad \mathcal{O}(L^2 d_h). \end{aligned}$$

Summing over all h heads:

$$T_{\text{score}} = \mathcal{O}\big(h\big(L d_h^2 + L^2 d_h\big)\big) = \mathcal{O}\left(\frac{L d^2}{h} + L^2 d\right). \tag{49}$$

Stage 4: Softmax. Row-wise softmax over h score matrices of size $L \times L$:

$$T_{\text{softmax}} = \mathcal{O}(hL^2). \tag{50}$$

Stage 5: Value projection and output. This stage is structurally identical to standard MHA. Value projections $V_i = XW_V^{(i)}$ cost $\mathcal{O}(Ldd_v)$ per head; the attention-weighted sum $A_i V_i$ costs $\mathcal{O}(L^2 d_v)$ per head; and the output projection W_O costs $\mathcal{O}(Ld^2)$. With $d_v = d_h$ and summing over heads:

$$T_{\text{value}} = \mathcal{O}(Ld^2 + L^2 d). \tag{51}$$

Total per-layer cost of DAMCHA. Combining the five stages, and writing $T_{f_\theta}^{\text{amort}} = T_{f_\theta}/N$ for the shared generator:

$$T_{\text{DAMCHA}}^{\text{layer}} = \mathcal{O}\left(\frac{L d^2}{h} + L^2 d\right) + \frac{1}{N}\,\mathcal{O}\big(Ld + Pw^2 + wd^2\big)\,. \tag{52}$$

D.3 Comparison with Standard MHA

Standard MHA per layer involves:

$$\begin{aligned} \text{Q, K projections (per head, } h \text{ heads}): &\quad \mathcal{O}(Ld \cdot d_h \cdot 2h) = \mathcal{O}(Ld^2), \\ \text{V projection}: &\quad \mathcal{O}(Ld^2), \\ QK^\top \text{ (per head)}: &\quad \mathcal{O}(L^2 d_h \cdot h) = \mathcal{O}(L^2 d), \\ \text{Softmax}: &\quad \mathcal{O}(hL^2), \\ \text{Output projection}: &\quad \mathcal{O}(Ld^2). \end{aligned}$$

$$T_{\text{MHA}}^{\text{layer}} = \mathcal{O}(Ld^2 + L^2 d). \tag{53}$$

Table 7 summarises the comparison. Three observations follow directly from this comparison.

Score computation is cheaper. The dominant cost of attention score formation in MHA is $\mathcal{O}(Ld^2)$ (from Q, K projections plus $QK^\top$). In DAMCHA, the corresponding cost is $\mathcal{O}(Ld^2/h)$, a factor-of-h saving, because f_θ directly outputs the effective metric $m_i(X)$ without the $d \to d_h$ projection step.

Overhead of f_θ is amortised. Under stack-wise parameter sharing, f_θ is invoked once per forward pass regardless of N. As N grows (the typical setting for deep Transformers), its per-layer contribution tends to zero.

Table 7: Per-layer time complexity: DAMCHA vs. standard MHA. † indicates the stage is shared across all N layers and amortises to $\mathcal{O}(\cdot)/N$.

Stage	MHA	DAMCHA
Q, K projections	$\mathcal{O}(Ld^2)$	— (replaced by f_θ)
Metric generator f_θ†	—	$\mathcal{O}(Ld + Pw^2 + wd^2)$
RowSum aggregation	—	$\mathcal{O}(d^2)$
Attention scores	$\mathcal{O}(Ld^2 + L^2d)$	$\mathcal{O}(Ld^2/h + L^2d)$
Softmax	$\mathcal{O}(hL^2)$	$\mathcal{O}(hL^2)$
V projection + output	$\mathcal{O}(Ld^2 + L^2d)$	$\mathcal{O}(Ld^2 + L^2d)$
Total (per layer)	$\mathcal{O}(Ld^2 + L^2d)$	$\mathcal{O}(Ld^2/h + L^2d)$ + amort. f_θ

D.4 Parameter Count

Let $\Theta(\cdot)$ denote the exact parameter count.

Standard MHA. Each layer contains projections $W_Q^{(i)}, W_K^{(i)}, W_V^{(i)} \in \mathbb{R}^{d\times d_h}$ for all h heads and an output projection $W_O \in \mathbb{R}^{d\times d}$:

$$\Theta_{\text{MHA}}^{\text{layer}} = 3h \cdot d \cdot d_h + d^2 = 3d^2 + d^2 = 4d^2. \tag{54}$$

For an N-layer stack: $\Theta_{\text{MHA}} = 4Nd^2$.

DAMCHA. Each layer retains $W_V^{(i)}$ and W_O but discards $W_Q^{(i)}$ and $W_K^{(i)}$, which are superseded by f_θ:

$$\Theta_{\text{DAMCHA}}^{\text{layer}} = h \cdot d \cdot d_h + d^2 = 2d^2. \tag{55}$$

The shared metric generator contributes a *single* parameter block:

$$\Theta_{f_\theta} = d \cdot w + (P-2)w^2 + w \cdot d^2 = \mathcal{O}(Pw^2 + wd^2). \tag{56}$$

Total for the N-layer stack:

$$\Theta_{\text{DAMCHA}} = 2Nd^2 + \Theta_{f_\theta}. \tag{57}$$

Net saving.

$$\Delta\Theta = \Theta_{\text{MHA}} - \Theta_{\text{DAMCHA}} = 2Nd^2 - \Theta_{f_\theta}. \tag{58}$$

This saving is strictly positive whenever $2Nd^2 > \Theta_{f_\theta}$, i.e. $N > \Theta_{f_\theta}/(2d^2) \approx Pw^2/(2d^2) + w/2$. For the GPT-2 scale configuration ($N = 12$, $d = 768$, $w = 1536$, $P = 9$) the right-hand side evaluates to approximately 12.8, so DAMCHA has essentially the same parameter count as MHA at this scale and fewer parameters for larger N. Concretely, DAMCHA achieves a $\sim 25\%$ reduction in attention-module parameters relative to MHA when $N \geq 15$ and $w \approx 2d$.

E Additional Distribution Plots for Sample Generation

Figures 4–7 show that DAMCHA consistently changes the *shape* of the generation-score distributions rather than only improving the average. Across datasets, DAMCHA tends to shift probability mass toward the high-score region and reduce the density of low-score samples, indicating fewer low-quality outliers. In addition, the KDE curves under DAMCHA are typically more concentrated, suggesting improved stability (smaller dispersion) in generation quality. These distribution-level patterns align with the main-text quantitative results and provide evidence that the gains are not driven by a small subset of samples.

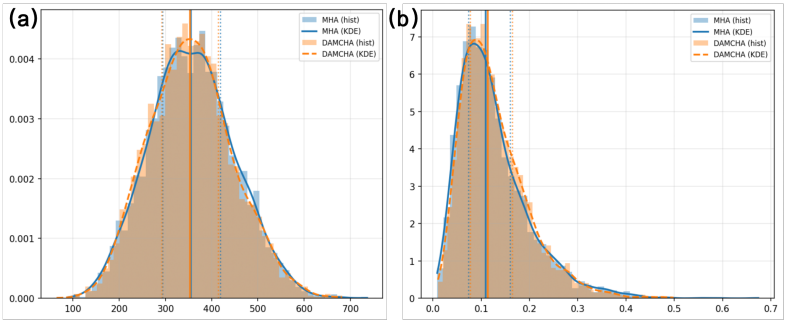

Figure 6: Supplementary distribution plot on CIFAR-100.

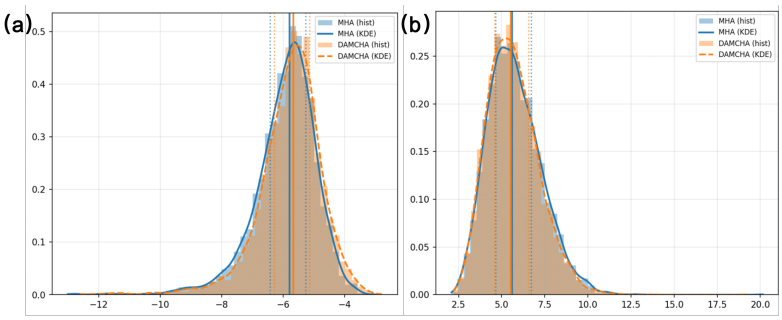

Figure 7: Supplementary distribution plot on CommonGen.

F The Evolution of Cross-Head Metric Matrix during training

Figure 8 visualizes the learned cross-head metric matrices $M(X)$ at different training stages on CIFAR-10. We show four representative layers (layer-0, 1, 2, 3) at epoch 1 (top row) and epoch 50 (bottom row). At epoch 1, the matrices exhibit relatively uniform structures with weak off-diagonal interactions. By epoch 50, three distinct patterns emerge: **(1) Strong diagonal blocks:** The diagonal 8×8 blocks corresponding to $M_{i,i}(X)$ show high activation values, indicating well-learned intra-head similarities that capture head-specific feature patterns. **(2) Structured off-diagonal blocks:** Systematic cross-head interactions appear in the off-diagonal regions, particularly in middle layers (layer-1, layer-2). These blocks encode how queries from one head measure similarity under the feature patterns of another head, implementing the cross-head cooperation mechanism. **(3) Layer-specific specialization:** Early layers (layer-0, layer-1) develop denser and more diverse off-diagonal structures, suggesting active cross-modal feature integration in shallow representations. Deeper layers (layer-2, layer-3) maintain sparser but more targeted interactions, focusing on task-specific head coordination. This hierarchical specialization aligns with the intuition that early layers perform general feature extraction while deeper layers implement task-specific reasoning.

G Gradient Flow Analysis for Stack-Wise Sharing

To understand how parameter sharing stabilizes training, Figure 9 visualizes the gradient contribution heatmaps of the cross-head metric matrix under two settings: (a) without sharing and (b) with the standard DAMCHA sharing mechanism. Each heatmap shows the L_2 norm of gradients flowing into each block of the metric matrix $M \in \mathbb{R}^{hd \times hd}$ during training.

Without sharing (panel a), gradients are distributed unevenly across the matrix, with some blocks receiving disproportionately large updates (bright regions) while others remain nearly inactive (dark regions). This uneven flow leads to unbalanced learning where certain cross-head interactions dominate while others fail to develop. In contrast, with parameter sharing (panel b), gradients distribute more uniformly across the entire matrix. The diagonal blocks maintain strong but controlled gradients, while off-diagonal blocks show moderate and consistent activation. This balanced gradient

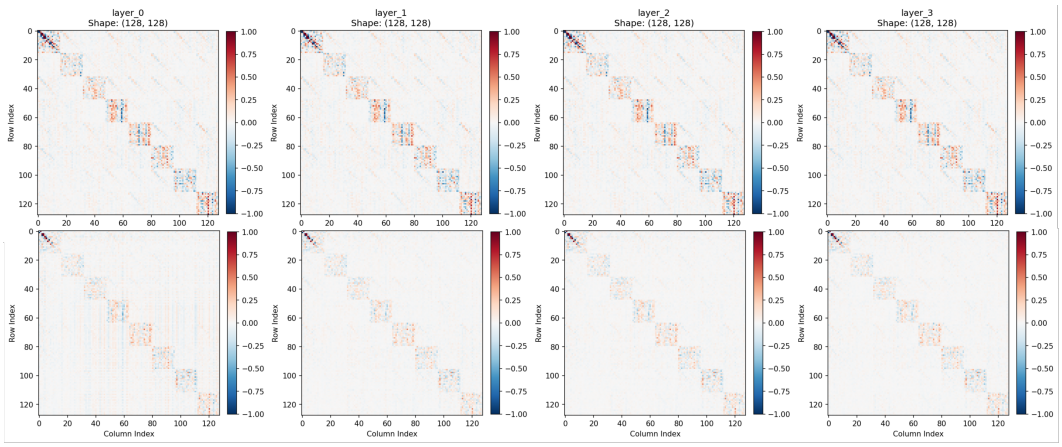

Figure 8: Cross-head metric matrix visualization on CIFAR-10. The top row shows the state of the cross-head metric matrix at epoch 1, and the bottom row shows the state at epoch 50.

flow prevents any single layer from learning degenerate or overly specialized metrics, promoting cooperative learning across layers.

The visualization explains the performance gap in Table 4: shared parameters act as an implicit regularizer, forcing all layers to reuse a consistent family of metric transformations. This constraint prevents layer-specific overfitting and encourages the model to learn generalizable cross-head interaction patterns. Qualitative contour-map comparisons across epochs are included in Appendix H (Fig. 10), supporting the ablation findings.

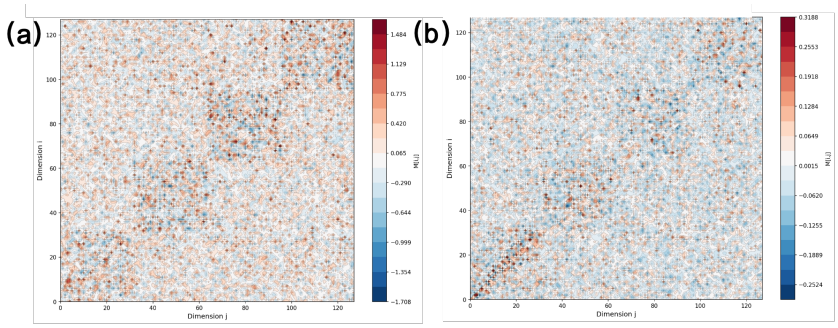

Figure 9: (a) Gradient contribution heatmap of the cross-head metric matrix when the sharing mechanism is disabled. (b) Gradient contribution heatmap of the cross-head metric matrix under the standard DAMCHA setting.

H Additional Visualizations for Ablation and Metric Geometry

Figure 10 provides a qualitative ablation comparison across training epochs for WMT dataset. The w/o $f(x)$ variant tends to exhibit less coherent contour structures and slower stabilization over epochs, whereas DAMCHA forms clearer and more stable geometric patterns as training progresses. This supports the view that the input-adaptive generator $f(x)$ helps shape a more structured and stable metric geometry.

Figure 11 further corroborates this observation from the perspective of the learned metric matrix M. Compared with the static design (w/o $f(x)$), the DAMCHA surface typically appears more organized (with smoother variations and fewer irregular spikes), consistent with a better-conditioned metric landscape.

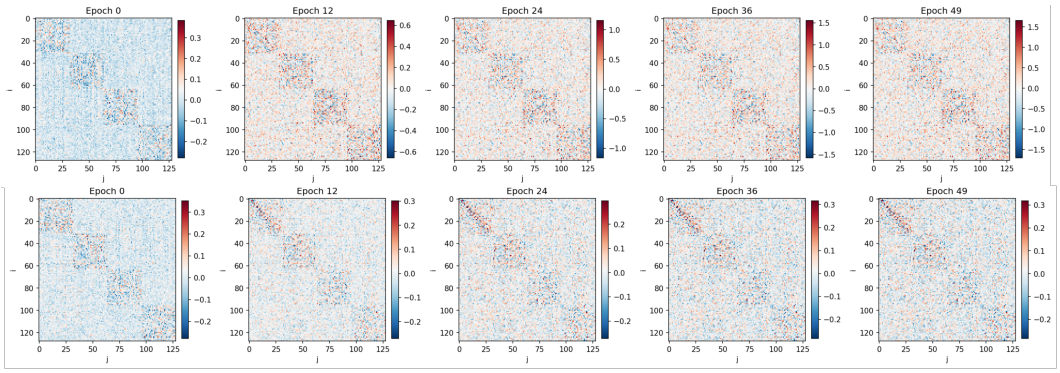

Figure 10: Contour-map visualizations across training epochs (Epoch 0/12/24/36/49). Top: w/o $f(x)$; Bottom: DAMCHA.

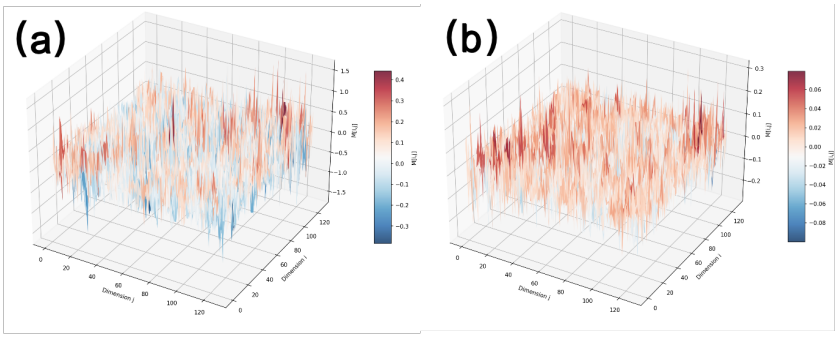

Figure 11: 3D surface visualization of the learned metric matrix M. Left: w/o $f(x)$; Right: DAMCHA.

I Synthetic Tasks

In this section, we design two synthetic tasks (regression and classification) to more explicitly present the advantages and underlying mechanism of our cross-head metric learning framework: modeling dependencies that require coordinated attention across multiple subspaces.

We consider a feature space formed by 4 subspaces. Each subspace is constructed by a vector-valued variable, respectively, denoted by $\{\boldsymbol{x}_i\}_{i=1}^4$. Each $\boldsymbol{x}_i \in \mathbb{R}^{d_e}$ and its data is generated following $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We define the following 4-dimensional regression system

$$y_i = w_i^\top \boldsymbol{x}_i + \lambda \sum_{j \neq i} (\boldsymbol{x}_i^\top W_{ij} \boldsymbol{x}_j), \tag{59}$$

where $w_i \in \mathbb{R}^{d_e}$ is a within-subspace weight vector, $W_{ij} \in \mathbb{R}^{d_e \times d_e}$ is a cross-subspace interaction matrix. All the weight parameters are initialized by sampling from $\mathcal{N}(0, 0.25)$. When $\lambda = 0$, the task reduces to independent within-subspace prediction. As λ increases, cross-subspace interactions become more critical, and DAMCHA's cross-head metric can provide advantages.

For a C-classes problem, we define a target classification system. The score for class c is computed as:

$$\text{score}_c = \sum_{i<j} (x_i^\top W_{ij}^c x_j), \tag{60}$$

where W_{ij}^c is a class-specific cross-subspace interaction matrix. W_{ij}^c is constructed as: $W_{ij}^c = \text{scale} \cdot U \odot V$ where $U, V \in \mathbb{R}^{d_e \times d_e}$ are randomly initialized and $\text{Rank}(W_{ij}^c) = d_e/2$. The final label is determined by $\text{label} = \arg\max_c(\text{score}_c + \text{noise})$ where noise is added for regularization. λ controls the entanglement level (strength of cross-subspace dependencies).

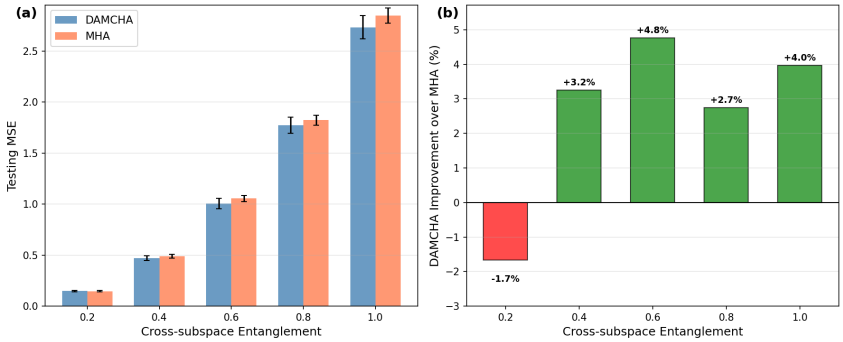

Figure 12: Cross-subspace regression: DAMCHA vs. MHA. (a) comparison results (testing MSE) under different entanglement levels. (b) DAMCHA's relative advantage under different entanglement levels.

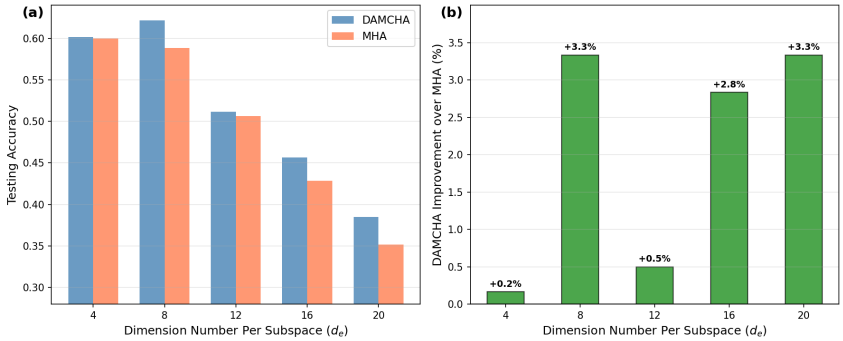

Figure 13: Cross-subspace classification: DAMCHA vs. MHA. (a) Comparison results (accuracy %) under different d_e. (b) Relative improvement of DAMCHA over MHA across different d_e.

For both systems, we train two 4-head Transformer models with head dimension 32. For the regression system, under $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, the model is trained for 10K steps with batch size 64. The model for the classification system is trained with 14K steps and batch size 128.

For regression task, we evaluate DAMCHA and MHA using testing mean square error (MSE). Figure 12(a) presents comparison results with error bars, which demonstrates clear advantages of DAMCHA. Figure 12(b) reveals a more nuanced picture by plotting DAMCHA's relative advantage over MHA. At $\lambda = 0.2$, DAMCHA underperforms MHA by 1.7%, when only limited cross-subspace interaction is needed, As λ increases from 0.2 to 1, DAMCHA's advantage grows from -1.7% to $+4.8\%$, confirming that the benefit scales with the strength of subspace entanglement.

For classification task, for comprehensive and robust evaluation, we vary $d_e \in \{4, 8, 12, 16, 20\}$, resulting in total dimensions from 16 to 80. Figure 13(a) provides accuracy comparison against d_e, visually confirming DAMCHA's consistent advantage across all settings. It also reveals an important scaling property: as d_e increases, both methods degrade, but DAMCHA degrades more slowly. This implies that DAMCHA's cross-head coordination becomes increasingly valuable in high-dimensional settings where MHA's independent heads cannot effectively capture the full feature space interactions. Figure 13(b) quantifies the relative improvement, further illustrating the advantages of DAMCHA.

The two synthetic experiments identify clear conditions under which DAMCHA provides advantage: (i) when the target dependencies explicitly involve interactions across feature subspaces that are assigned to different attention heads, (ii) such cross-subspace interactions are sufficiently strong (e.g., $\lambda \geq 0.2$ in the regression system), and (iii) the feature dimensionality is moderate to high (e.g., $d_e \geq 8$ in the classification system), where richer representational capacity becomes critical. These controlled findings offer a principled explanation for DAMCHA's strong empirical performance on real-world benchmarks, where complex, multi-faceted dependencies naturally arise. In such settings, the ability to explicitly model cross-head interactions allows DAMCHA to capture dependencies. When the underlying data-generating process is relatively simple, or when meaningful dependencies can be captured within individual heads without requiring cross-head coordination, standard MHA

may already be sufficient, leading to comparable generalization performance. This observation highlights that DAMCHA is most beneficial in scenarios where cross-subspace interactions are intrinsic to the task, while remaining competitive in simpler regimes.

J Computational Resources and Runtime Details

All experiments in this paper were conducted on a single server with the following hardware and software configuration:

- **GPU:** NVIDIA RTX 3090 (24 GB VRAM) $\times$ 1
- **CPU:** 12 vCPU Intel(R) Xeon(R) Platinum 8255C @ 2.50 GHz
- **Operating System:** Ubuntu 22.04
- **Python:** 3.12
- **PyTorch:** 2.7.0
- **CUDA:** 12.8

Per-Experiment Runtime. Table 8 summarizes the approximate wall-clock training time for each experimental setting.

Table 8: Approximate training time per experimental setting. All runs use a single RTX 3090 GPU.

Experiment	Dataset	Steps / Epochs	Approx. Training Time
Decoder-only Transformer	CIFAR-10	50 epochs	$\sim$3 h
	CIFAR-100	50 epochs	$\sim$3 h
	WMT14	50 epochs	$\sim$5 h
	CommonGen	50 epochs	$\sim$4 h
GPT-2 Pretraining	Enwik8	100K steps	$\sim$8 h
	Text8	100K steps	$\sim$8 h
	Mix	100K steps	$\sim$10 h
ViT-S/16 Classification	ImageNet-1K	500 epochs	$\sim$72 h
Synthetic Tasks	Regression / Classification	10K–14K steps	$<$30 min

K Comparison with Baseline Methods

Mechanistic Advantages over Competing Attention Methods. From a theoretical standpoint, the advantages of DAMCHA over all baseline methods can be understood through three independent but complementary mechanisms: the *structural expansion* of the metric feasible class, the *dynamic adaptivity* of the learned similarity geometry, and the *rank enhancement* induced by cross-head aggregation. No existing baseline addresses all three simultaneously, and the gap between DAMCHA and each competitor can be precisely traced back to which of these mechanisms it lacks. At the empirical level, these distinctions consistently translate into superior performance across all benchmark tasks, stemming from two fundamental innovations absent in every prior method: *(i)* learning the full cross-head metric matrix $\mathbf{M}(X)$ that explicitly captures inter-head interactions beyond the block-diagonal constraint of standard MHA, and *(ii)* parameterizing this metric as an input-dependent function, enabling the attention geometry to dynamically adapt to the local manifold structure of the data at inference time.

vs. MHA. Standard MHA imposes a block-diagonal restriction on the joint head-space metric, structurally zeroing out all off-diagonal interactions, fixing the metric after training, and capping the per-head rank at the projection dimension, thereby simultaneously introducing all three deficiencies that DAMCHA is designed to resolve. DAMCHA strictly generalizes MHA: MHA is exactly recoverable as a special case by setting all off-diagonal metric blocks to zero and using a constant generator, while the converse is provably false ($\mathcal{F}_{\text{MHA}} \subsetneq \mathcal{F}_{\text{DAMCHA}}$). The resulting performance gap decomposes into two independent additive components: a structural gap from the enlarged metric feasible class, and a dynamic gap from input-adaptive metric generation, both of which are strictly

positive under mild conditions on the input distribution. Concretely, MHA's static metric can at best approximate the average attention geometry $\hat{M} \approx \mathbb{E}_X[M^*(X)]$, whereas DAMCHA's adaptive generator f_θ targets the pointwise optimum $M^*(X)$ for every input, translating to a 31.1% relative reduction in FID@Top-50 on CIFAR-10 and a 35.7% improvement in BARTScore on WMT.

vs. THA. Talking-Heads Attention inserts learned linear projections across the head dimension immediately before and after the softmax, allowing attention weights to be redistributed across heads at the *weight* level. However, this operates entirely at the *post-hoc aggregation* level: the underlying per-head similarity metric $M_{i,i} = W_Q^{(i)}(W_K^{(i)})^\top$ remains unchanged, static, and rank-limited. From a metric-learning perspective, THA never modifies the similarity geometry itself; its cross-head interaction is confined to a linear remixing of already-formed attention patterns rather than a genuine coupling at the similarity-formation stage, and THA therefore inherits all three limitations of MHA in terms of metric expressiveness. DAMCHA, by contrast, intervenes directly at the metric level before any attention score is computed, and this structural distinction is reflected empirically: THA yields worse MSE, FID, and NLL than DAMCHA across all datasets while requiring substantially more memory (255.40 vs. 204.70 on CIFAR-10) and higher latency.

vs. DCMHA. DCMHA introduces input-dependent composition of attention score and weight matrices via a Compose function, making it the baseline that most closely approaches DAMCHA's dynamic adaptivity. The key theoretical distinction, however, is *where* the input-dependence is introduced: DCMHA's Compose function operates on scores already derived from fixed, low-rank per-head bilinear forms, reweighting and mixing existing scores but never escaping the rank ceiling $\mathrm{rank}(M_{i,i}) \leq d_k$ imposed by the individual head projections. DAMCHA's adaptivity is applied directly to the metric that *generates* the scores, meaning the attention geometry itself is data-dependent before any score is produced. DCMHA can be understood as composing within a restricted space of attention patterns, whereas DAMCHA reconfigures the similarity space from which all patterns are drawn. The structural advantage of expressing full-rank, off-diagonal cross-head metrics therefore persists even when both methods employ similarly expressive input-dependent generators. Experimentally, DAMCHA outperforms DCMHA on all metrics in both the generation experiments and the GPT-2 pretraining experiments, with cross-entropy improvements of 1.90%–3.70% across Enwik8, Text8, and Mix, while also using less memory (204.70 vs. 292.71 on CIFAR-10) and achieving lower latency.

vs. ColMHA. Collaborative MHA addresses head redundancy by sharing key/query projections via Tucker tensor decomposition, providing a compact reparametrization. From a theoretical standpoint, however, the resulting model is *entirely static*: the shared projections are determined at training time and carry no mechanism for input-dependent adaptation, keeping the metric class within a subset of fixed bilinear forms. Moreover, the relief from the low-rank bottleneck that ColMHA obtains is a post-hoc algebraic correction applied to a pre-trained model, not a structural property of the architecture itself. DAMCHA achieves rank enhancement organically through the RowSum aggregation of cross-head metric blocks (with no reparametrization required) and simultaneously endows this richer metric class with input-adaptive generation from the outset. On both image and text generation tasks, DAMCHA achieves substantially better FID@Top-50 and BARTScore@Top-50 than ColMHA, confirming that static projection sharing cannot substitute for dynamic cross-head metric learning.

vs. MoA. Mixture of Attention Heads introduces token-level selectivity by routing each token to a sparse subset of heads via a learned gating network, providing a limited form of input-dependent adaptivity. However, routing operates at the *output* level by sparsely combining independently formed, low-rank attention distributions; each selected head still computes an independent Euclidean dot-product similarity within its own subspace, and there is no path by which the similarity geometry of one head can be influenced by or coordinated with that of another at the score-formation stage. MoA therefore provides no mechanism for modeling metric-level interactions between heads, which is precisely the coupling that DAMCHA encodes as a first-class structural property. This theoretical limitation is compounded by a practical one: the sparsity required by MoA incurs the highest memory footprint of all compared methods (321.07 on CIFAR-10 vs. 204.70 for DAMCHA) and the highest GFLOPs (0.1101 vs. 0.0786), without commensurate performance gains.

vs. MMA. Mixed Multi-Head Attention enforces head diversity by assigning fixed attention masks (global, local, forward, and backward) to different heads. From a theoretical perspective, this is a strict *special case* of MHA: the additional structural constraints on admissible attention patterns

further *reduce* the effective function class rather than expanding it, making MMA theoretically weaker than standard MHA in terms of metric expressiveness. MMA's diversity is entirely predetermined and input-independent, and it does not interact with the similarity metric at all; it merely restricts which positions each head is permitted to attend to, without altering the underlying dot-product geometry. This rigidity becomes a critical liability on tasks requiring fine-grained, context-sensitive head coordination: while MMA achieves competitive NLL on WMT, it degrades significantly on BARTScore@Top-50 and collapses on CommonGen, precisely the settings where DAMCHA's adaptive cross-head metric excels.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state DAMCHA's two core innovations: learning the full cross-head metric matrix and parameterizing it as an input-dependent function. These claims are directly supported by the theoretical proofs, ablation studies, and empirical experiments across image generation, text generation, language modeling, and classification provided in the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: DAMCHA introduces a learnable metric generator f_θ shared across all layers, which brings a new hyperparameter — the capacity (depth) of f_θ — that may require task-specific tuning, as performance improves consistently with deeper generators across all evaluated datasets. Additionally, the stack-wise parameter sharing constrains all layers to reuse the same family of metric transformations, which may limit layer-specific specialization in very deep or highly heterogeneous architectures. These trade-offs are discussed in the ablation study.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The strict expressiveness result ($\mathcal{S}_{\mathrm{MHA}} \subsetneq \mathcal{S}_{\mathrm{DAMCHA}}$) and the dynamic approximation advantage theorem are both proved in full in the appendix. All assumptions are explicitly stated prior to each result. The expressiveness proof proceeds via explicit construction for containment and a rank-based counterexample for strictness.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All key hyperparameters are reported for each experimental setting, including model architecture (number of layers, heads, model and head dimensions), optimizer (Adam), learning rates, batch sizes, training steps or epochs, and evaluation metrics. Hardware and software configurations are documented in the appendix. An anonymous code repository is linked in the introduction.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository is provided and linked in the introduction. All datasets used are publicly available standard benchmarks, with references and URLs provided in the main text and footnotes.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

Justification: Training details including architecture configuration, optimizer type, learning rates, batch sizes, number of training steps or epochs, and evaluation metrics are reported for all experimental settings in the main text. Ablation configurations and synthetic task settings are detailed in the ablation study and the appendix respectively.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are reported in the synthetic regression and classification experiments in the appendix. However, the main generation and language modeling results do not include error bars, as running multiple full training runs across all methods and datasets on a single GPU would be computationally prohibitive given the hardware constraints described in the appendix.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix documents the full hardware and software configuration (NVIDIA RTX 3090 with 24 GB VRAM, 12 vCPU Intel Xeon Platinum 8255C @ 2.50 GHz, Ubuntu 22.04, Python 3.12, PyTorch 2.7.0, CUDA 12.8) and provides approximate wall-clock training times for each experimental setting. Per-method memory usage, GFLOPs, and latency are also reported in the efficiency comparison table.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper proposes a general-purpose attention mechanism for Transformer architectures. It does not involve human subjects, sensitive data collection, or applications with direct negative societal impact. All datasets used are standard publicly available benchmarks. The work conforms fully with the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents foundational research on attention mechanisms. The proposed method improves general-purpose representation learning and does not introduce new capabilities with a direct path to negative applications beyond those already associated with standard Transformer models. The societal impacts are consistent with those of general machine learning research, as noted in the Impact Statement.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes an attention mechanism and does not release a large pretrained model or scraped dataset. All experiments use standard publicly available benchmarks. No high-risk model release is involved, and no specific safeguards are required.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and baseline methods are properly cited with their original publications. Dataset URLs are provided in footnotes where applicable. All baseline methods (THA, DCMHA, ColMHA, MoA, MMA) are cited with their original papers. The GPT-2 and ViT architectures are used in standard publicly available configurations.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The DAMCHA attention mechanism is fully described in the method section, covering the cross-head metric matrix formulation, adaptive learning procedure, attention computation, and the stack-wise sharing mechanism. The end-to-end data flow is further illustrated in the appendix. The anonymous code repository contains all training scripts and model configurations.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve any crowdsourcing or research with human subjects. All experiments are conducted on standard machine learning benchmarks using automated evaluation metrics.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve any research with human subjects. No IRB approval is required.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A]

    Justification: LLMs are not used as a core methodological component in this research. DAMCHA is evaluated on GPT-2-scale language models as a drop-in attention replacement, but the core contribution is the proposed attention mechanism itself. Any use of LLMs was limited to writing assistance and does not impact the core methodology, scientific rigor, or originality of the research.