arXiv:2305.04800v2 [cs.LG] 3 Aug 2023

Mlinear：重新思考用于时间序列预测的线性模型

Wei Li[1], Xiangxu Meng[1], Chuhao Chen[1] 和 Jianing Chen[1⋆]

哈尔滨工程大学计算机科学与技术学院，中国哈尔滨 150001{wei.li、mxx、chenchuhao、kaqiz}@hrbeu.edu.cn

摘要。 近年来，时间序列预测研究取得了显著进展，研究重点逐渐转向分析时间序列数据的性质，例如通道独立性（CI）和通道依赖（CD），而不再仅仅关注设计复杂的预测模型。然而，当前研究主要孤立地关注 CI 或 CD，如何有效结合这两种相互对立的性质以实现协同效应，仍是一个尚未解决的问题。本文仔细考察了 CI 和 CD 的对立性质，并提出了一个尚未得到有效回答的实际问题，即：“如何有效混合时间序列的 CI 和 CD 性质，以获得更好的预测性能？”为回答这一问题，我们提出了 Mlinear（MIX-Linear），一种主要基于线性层的简单而有效的方法。Mlinear 的设计理念主要包括两个方面：（1）根据不同输入时间序列的时间语义动态调整 CI 和 CD 性质；（2）提供深度监督，以调整“CI 预测器”和“CD 预测器”的个体性能。此外，在实证研究中，我们引入了一种新的损失函数，在多个数据集上显著优于广泛使用的均方误差（MSE）。在涵盖多个领域且广泛使用的时间序列数据集上进行的实验表明，在序列输入相同（336 或 512）的情况下，就 7 个数据集上的 MSE 和平均绝对误差（MAE）评价指标而言，我们的方法优于 PatchTST（最新的基于 Transformer 的方法）。具体而言，在序列长度为 336 的输入下，我们的方法以 21:3 的比例显著优于 PatchTST；在序列长度为 512 的输入下，该比例为 29:10。此外，综合考虑训练时间和推理时间，我们的方法在单元级别具有 $10\times$ 效率优势。

关键词： 时间序列预测、深度学习、线性模型、通道独立性、通道依赖性

1 引言

通过分析历史数据来预测未来趋势，是一个经典的研究领域，在工业界和学术界都具有重要的应用价值。作为一个热门课题

⋆ 通讯作者

近年来，作为时间序列预测中的热门方向，多变量预测受到了广泛关注，并取得了显著进展。在过去几十年里，时间序列预测（TSF）方法经历了从 ARIMA [1], 等传统统计方法和 GBRT [10], 等机器学习技术，向基于深度学习的方法转变。此外，得益于基于深度学习的模型在知识容量方面的优势，序列预测逐渐拓展至更广泛的领域，例如长序列时间序列预测（LSTF）。尤其是，循环神经网络 [15],、时间卷积网络 [2] 和 Transformer [33] 等新技术已广泛应用于这一领域。

在众多优秀的长序列时间序列预测（LSTF）方法中，Transformer 无疑是最出色的一种。源自自然语言处理的基于 Transformer 的模型 [33] 在计算机视觉 [7, 12, 20–22、25], 图数据处理 [3, 26, 39], 以及各种下游任务 [17, 18, 30, 32, 37]中均取得了优异成果。Transformer 出色的全局建模能力使其能够关注长序列中数据点之间的长跨度关系，从而具备捕捉长期时间依赖关系的能力。这些基于 Transformer 的 LSTF 方法取得了重大进展，甚至在北京冬奥会期间实现了实际应用 [35]。据我们所知，这类方法中最具代表性的两种是 Informer 和 Autoformer。引用数量也可以从侧面印证这一点。Informer [42] 针对序列数据处理中活跃注意力呈长尾分布的问题，基于这一特性提出了一种稀疏注意力计算方法。这将时间复杂度从 $O(L^2)$ 降低至 $O(LlogL)$。另一方面，Autoformer 使用序列分解作为预处理方法，并以自相关机制取代点积注意力机制，实现了序列级连接，并将时间复杂度降低至 $O(LlogL)$。与此同时，还有一些其他优秀的基于 Transformer 的 LSTF 方法，尝试从不同角度实现准确的长期序列预测，例如频域注意力和语义注意力。Fedformer [43], 、Pyraformer [24], 和 Triformer [5]便是其中的代表性工作。

尽管基于Transformer的方法取得了显著的预测性能，但值得注意的是，大多数研究主要沿用了标准的编码器—解码器架构。虽然许多研究声称从语义和频域角度进行了改进，但仔细审视以往工作便会发现，这些改进主要集中在效率上，尤其是注意力机制的计算。换言之，这些令人瞩目的结果主要得益于多头注意力机制的全局建模能力，却未能深入分析时间序列的本质以提高预测准确性。近期研究出现了显著变化：将通道独立（CI）和通道依赖（CD）的思想引入时间序列预测后，一些工作仅使用简单的线性组合 [40] 和经典 Transformer [28]便取得了极为出色的性能。例如，

PatchTST [28]将CI的概念引入Transformer架构，并结合分块思想对局部时间依赖关系进行建模。PatchTST并未对Transformer架构进行重大改进，却仅凭经典的Transformer编码器，就取得了超越以往精心设计的Transformer工作的性能。相比PatchTST，Dlinear [40] 的表现或许更加令人惊讶。借助CI的概念，它们仅通过堆叠线性层，就以极低的计算成本，在基于Transformer的长序列时间序列预测方法上取得了显著进步。

深入理解多变量长时间序列预测中 CI 和 CD 的概念。一个包含 L 个时间单位和 N 个变量的时间序列 $L \in \mathbb{R}^{L \times N}$ 可以被视为由多个独立通道组成的集合，如图 B 所示。由于每个变量都来源于特定预测任务中的不同评价指标，因此它们之间存在隐藏且复杂的依赖关系。CD 是指将整个序列作为一个整体输入，对不同变量之间的依赖关系进行建模。相比之下，CI 是指将每个变量视为一个独立的序列，并专注于对每个变量内部的依赖关系进行建模。

尽管近期研究发现，在长序列时间序列预测任务中，通道独立相较于通道依赖具有显著优势，并放弃通道依赖以充分利用通道独立的特性，但我们认为这种做法并不可行，因为它完全忽视了通过通道依赖建模不同变量之间关系的可能性，导致信息瓶颈问题。此外，我们认为通道独立的显著优势只是特定结构带来的结果。如表 1 所示，对于我们提出的 Mlinear，通道依赖与通道独立之间并无显著的优势差异，而且二者在不同数据集上各有优势。因此，本文聚焦于通道独立与通道依赖之间的关系，并提出一个具有挑战性但切合实际的问题：如何有效融合通道独立与通道依赖，以兼得二者的优势？ 如表 1 所示，我们提出了一种简单的融合方案，并最终得到 Mlinear，显著提升了性能。

在本文中，我们旨在通过提出一种混合 CI 和 CD 的时间序列线性模型 MLinear 来回答这一问题，该模型包含三个关键设计组件：

混合多通道与单通道时间序列数据 传统上，在进行时间序列预测时，多变量（多通道）时间序列数据通常被视为一个整体输入，这已成为大多数基于Transformer的模型所采用的常规方法。由于多个变量之间存在相互依赖，近期研究通常将这种方法称为通道依赖（CD） [28,40], ，因为它体现了深度学习模型中不同通道之间的相互依赖。然而，由于不同通道的数据分布各异，CD可能导致数据污染和分布漂移。DLinear打破了这一做法，引入了多变量、多通道数据的通道独立（CI）预测。该方法将每个时间序列数据通道分别处理，颠覆了输入方式，并显著提升了性能。然而，尽管CI展现出良好的预测

方法	通道	依赖	CI	MIX	MLinear
Electricity	96	0.141	0.135	0.134	**0.133**
	192	0.154	0.150	**0.149**	**0.149**
	336	0.170	0.167	0.166	**0.165**
	720	0.209	0.206	0.205	**0.203**
ETTh1	96	0.364	0.372	0.367	**0.359**
	192	0.406	0.413	0.407	**0.400**
	336	0.429	0.433	0.428	**0.417**
	720	0.425	0.440	0.434	**0.429**
ETTm1	96	0.297	0.292	0.291	**0.289**
	192	0.335	0.335	0.334	**0.331**
	336	**0.369**	0.373	0.372	**0.369**
	720	**0.427**	0.431	0.430	0.428
计数		3	3	10	18

表 1. 展示了 CI、CD、线性聚合和 Mlinear 在三个数据集上的简要比较。所使用的损失函数是我们提出的损失函数。

性能，但尚未得到充分探索。此外，CD的物理意义也不应被完全忽视。

根据不同序列的时间语义动态调整 CI 和 CD一个自然的想法是混合 CI 和 CD 进行协同预测，但使用一组固定参数进行简单混合可能导致泛化能力不足，因为不同输入序列之间的时间依赖关系各不相同。例如，不同时间序列输入的时间语义有所差异。以峰值点为例，它们可能标志着一个周期的结束或开始；峰值点附近的子序列通常会在一段时间内持续保持较高值，而谷值点附近的峰值点则表现相反。为了解决这一问题，我们提出了一种混合方法，可根据不同输入的时间语义动态建模 CI 和 CD，同时将参数使用量和计算复杂度保持在极低水平。采用不同方法进行混合所产生的性能差异如表 1 所示。

针对 CI 和 CD 的深度监督直观而言，混合 CI 和 CD 就是结合这两个组件生成的预测结果，以获得更好的预测性能。通过提升两者各自的性能，我们可以获得更好的联合性能。深度监督能够有效监督更深层参数。因此，CI 和 CD 中不共享的深层参数可以得到更好的训练，从而使 CI 和 CD 各自的预测器表现更佳，进而提升联合性能。据我们所知，这是长序列时间序列预测领域首次尝试通过深度监督提升拟合性能。

更强大的损失函数在时间序列预测方法中，常见做法是针对不同模型使用均方误差（MSE）损失函数，例如 Informer、Autoformer、Fedformer、PatchTST 和 DLinear。然而，我们的实验表明，均方误差损失在时间序列预测领域表现不佳。大量研究表明，均方误差损失倾向于惩罚较大的数值，赋予它们更高的权重，而忽略较小数值的作用。如果样本中存在离群值，均方误差损失会为其赋予更高权重，但这是以牺牲其他正常数据点的预测性能为代价的，导致训练所得模型的整体性能下降。基于这一观察，我们提出了一种新的损失函数，与均方误差和平均绝对误差损失相比，它能显著提升模型的鲁棒性和预测性能。

我们将在以下章节中更详细地介绍 Mlinear 模型，并提供全面的实验结果，以确凿地支持我们的主张。我们通过预测结果和消融实验展示模型的有效性。

2 相关工作

近年来，基于学习的长序列时间序列预测（LSTF）方法取得了显著进展。从模型角度来看，基于Transformer的方法如雨后春笋般涌现，各种优秀的注意力机制和改进的架构层出不穷，并长期占据主导地位。然而，随着研究人员对通道独立（CI）和通道依赖（CD）特性的关注，非Transformer方法也获得了进一步的发展潜力。因此，我们将从这两个角度探讨相关LSTF方法的研究进展。此外，部分里程碑式工作的时间线如图1所示。

2.1 基于 Transformer 的长期时间序列预测。

近年来，基于 Transformer 的长序列时间序列预测（LSTF）方法因其备受赞誉的“全局建模”能力而受到越来越多的关注。自注意力机制使模型能够自动捕捉时间序列数据中不同时间步之间的依赖关系，从而帮助 Transformer 模型更好地理解时间序列数据中固有的时间依赖关系。

尽管 Informer [42] 并不是最早提出的模型，但它仍被认为是其中最值得注意的模型。Informer 尤为出色之处在于，它探索了全局注意力机制中的长尾分布问题，并由此提出了稀疏注意力机制。稀疏自注意力分析时间序列数据中时间依赖关系的稀疏性，并通过少量采样计算获得具有显著注意力分数的查询矩阵。通过使用数量有限的活跃查询进行 $query \times key$ 乘法运算，自注意力机制的时间复杂度从$O(L^2)$降至 $O(LlogL)$。Autoformer [35] 引入了一种分解架构

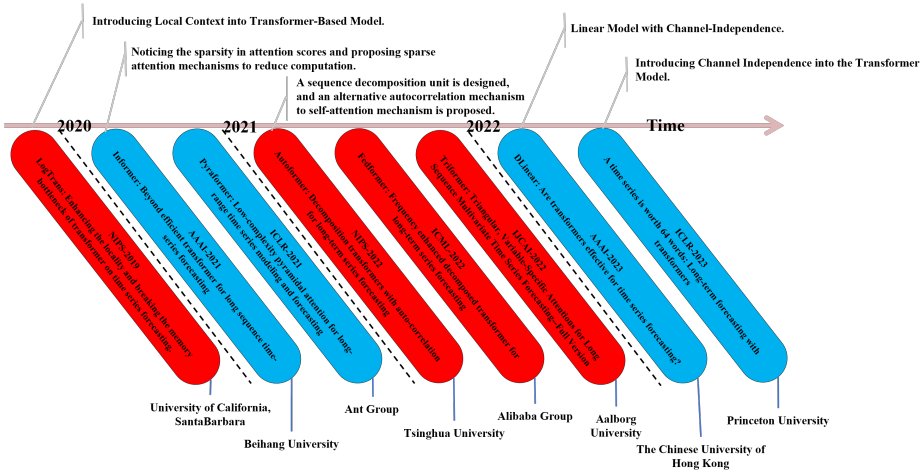


图 1. 时间序列预测里程碑时间线。

该架构能够从复杂的时间模式中提取出更多可预测的组件。基于随机过程理论，Autoformer 提出了自相关机制，以替代点对点注意力机制，实现序列级连接，并将时间复杂度降至 $O(LlogL)$，突破了稀疏注意力机制在信息利用方面的瓶颈。Fedformer [43]利用了大多数时间序列在知名基底（如傅里叶变换）下往往具有稀疏表示这一事实，并提出了增强频率 Transformer，以提升长期时间序列预测的性能。Pyraformer [24]采用带有跨尺度和尺度内连接的金字塔注意力模块，实现了线性复杂度。最新的基于 Transformer 的长序列时间序列预测工作 PatchTST [28], 将时间序列的通道独立属性与分块思想相结合，聚焦于时间子集进行建模，并取得了显著的预测性能。上述工作均属于通用领域；此外，Transformer 在特定领域的长序列建模方面也展现出了能力，例如股票趋势预测 [6,27,41] 和交通流量预测 [4,36,38]。总体而言，这些模型展示了 Transformer 架构应对时间序列预测挑战的有效性，并为该领域未来的研究开辟了令人振奋的新可能。

2.2 基于非 Transformer 的长期时间序列预测。

据我们所知，最早专注于时间序列预测的研究可追溯至 ARIMA [1], ，它遵循马尔可夫过程，并建立了用于递归序列预测的自回归模型。该模型非常简单，只需使用内生变量，无需其他外生变量。然而，自回归过程不足以处理非线性

和非平稳序列。此外，它们要求时间序列数据保持平稳，或经差分后达到平稳。因此，ARIMA 只能捕捉线性关系，无法捕捉非线性关系。

随着神经网络的快速发展，以及它们在其他领域展现出的卓越能力，基于循环神经网络的方法 [8, 14, 16, 29] 也应运而生，这得益于其捕捉输入序列与输出序列之间序列依赖关系、灵活处理可变长度序列以及对复杂模式进行建模的能力。例如，GCAR [11] 是一种专为多变量时间序列预测设计的模型，其创新之处在于通过可靠的交互增强对关键特征的记忆，并捕捉随时间变化的模式。该模型采用特征级和多层注意力机制，提取多头时间相关性，并区分外部特征的贡献。此外，GCAR 使用两个并行的长短期记忆网络来捕捉不同序列中连续的动态变化，并通过融合门平衡信息冲突。与传统的循环神经网络模型不同，时间卷积网络（TCN） [9, 13, 23, 34] 无需显式维护历史状态信息，而是通过卷积层捕捉时间序列数据的局部和全局依赖关系。例如，SCINet [23] 是一种基于时间卷积网络的时间序列预测方法，采用样本卷积与交互进行时间建模。它通过下采样—卷积—交互架构扩大卷积操作的感受野，并实现多分辨率分析，从而提升时间关系特征的提取效果和预测准确率。

上述方法尝试从统计学、循环神经网络和时间卷积网络的角度，为长期时间序列预测提供非 Transformer 的替代方案。然而，在涵盖广泛领域且序列长度较长的情况下，基于 Transformer 的长期时间序列预测方法在大多数情况下仍占主导地位。尽管如此，DLinear [40] 的出现撼动了 Transformer 家族：它将简单的线性层与通道独立特性相结合，以极低的计算成本超越了几乎所有基于 Transformer 的长期时间序列预测方法。这表明，非 Transformer 方法值得进一步探索。

3 方法论

3.1 通道独立和通道依赖的物理含义

通道依赖和通道独立都通过线性层进行序列建模，并且两种架构中使用的线性层具有相同的构成：输入大小等于序列长度，输出大小等于预测序列长度。它们的区别在于所采用的输入。每个通道独立线性层的输入 x_i 形状为 $\in \mathbb{R}^{1\times L}$，并且 $i = n$；而通道依赖线性层的输入X 形状为 $\in \mathbb{R}^{n\times L}$。在通道依赖中，线性层的权重在 n 序列之间共享，这意味着这些权重对不同通道之间的依赖关系进行建模。在通道独立中，有 n 个线性层，每个线性层都对一个

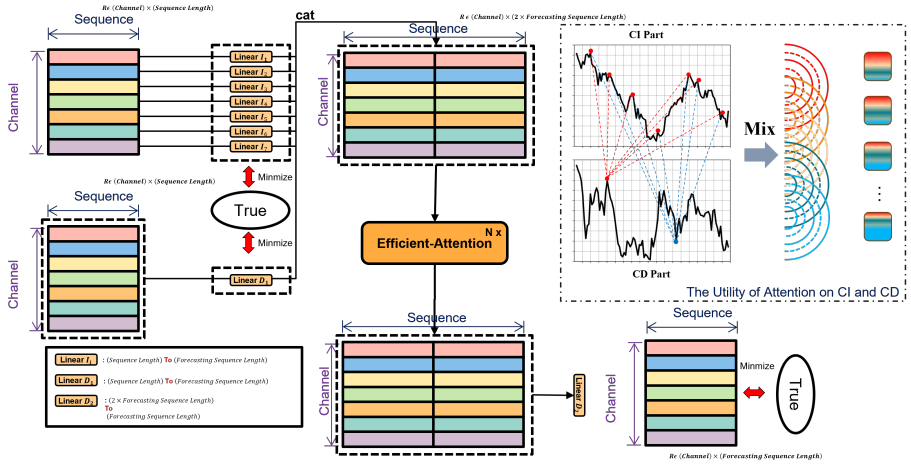


图 2. MLinear 的模型架构。

单通道序列进行建模，这意味着权重学习仅发生在单通道序列上。考虑输入 $X = [x_1, x_2, ..., x_n]$，通道独立和通道依赖的运算及拼接过程如下所示：

$$\begin{aligned} X_{\mathrm{CI}} &= \mathrm{Concat}(x_1 W_{\mathrm{CI}_1}, ..., x_i W_{\mathrm{CI}_i}), W_{\mathrm{CI}_i} \in \mathbb{R}^{L \times S} \\ X_{\mathrm{CD}} &= X W_{\mathrm{CD}}, W_{\mathrm{CD}} \in \mathbb{R}^{L \times S} \\ X_{\mathrm{CI+CD}} &= \mathrm{Concat}(X_{\mathrm{CI}}, X_{\mathrm{CD}}) \end{aligned} \tag{1}$$

其中， W_i 表示对应于 CI 和 CD 的线性权重。

3.2 时间语义感知的通道混合调制

为了避免使用固定权重对所有输入时间序列一视同仁，我们有必要动态调整 CI 和 CD 所提供的时间语义之间的关系，从而更恰当地聚合有助于预测的时间依赖关系。我们首先讨论固定权重与时间语义感知（TSA）权重之间的差异，然后展示所提出的方法如何调整不同的时间语义。

固定权重或TSA权重进行调制。以往大多数时间序列建模工作都基于Transformer或其变体。尽管这些不同工作的重点各不相同，但它们都拥有一个共同理念：对不同输入序列之间的依赖关系进行动态建模。这种动态建模理念不仅适用于时间序列预测任务，也适用于其他领域中的各种注意力机制。最新的时间序列长序列时间序列预测模型DLinear通过使用具有固定权重的简单线性层，取得了出色的性能

并结合通道独立（CI）的特性。然而，鉴于主要贡献在于通道独立（CI）带来的预测优势，在整体模型性能方面仍有进一步提升的空间。公式2和3展示了针对相同输入x，两种调优方法的不同过程。

$$X_{out} = X_{in} \odot \text{Attention}(WX_{in}) \tag{2}$$
$$X_{out} = WX_{in} \tag{3}$$

注意力意味着获取针对单个输入 X_{in}的时间语义而特定的系数（也称为注意力权重）。自注意力机制在长序列时间序列预测（LSTF）中得到了广泛应用。下一节中，我们将介绍利用注意力进行特殊语义调制的方法。我们的工作与众不同之处在于，我们同时对通道独立（CI）和通道依赖（CD）进行调制，并解决了使用固定权重进行简单调制所存在的局限性。此外，通过提供明确的证据，我们证明了注意力在长序列时间序列预测（LSTF）中的有效性，从而回应了 DLinear 关于注意力机制在长序列时间序列预测（LSTF）中作用的疑问。

为便于理解且不失一般性，我们考虑 CI 和 CD 的一种输入组合，记为 $X = [x_1, x_2, x_3] \in \mathbb{R}^{1\times 3}$。其中， x_1 和 x_2分别表示 CI 的输出， x_3 表示 CD 的输出。使用固定权重 $W \in \mathbb{R}^{3\times 3}$ 对 CI 和 CD 进行调优的过程如下：

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{3} w_{i1}x_i, \sum_{i=1}^{3} w_{i2}x_i, \sum_{i=1}^{3} w_{i3}x_i \end{bmatrix} \tag{4}$$

这种方法的缺点在于，由于权重 W 是固定的，不同 X 的组合模式保持不变。例如，考虑 $x_1 = 0.1$、 x_2 =0。3、 $x_3 = 0.6$ 以及聚合权重 $w_{11} = 1$、 $w_{22} = 2$、 $w_{33} = 3$。其总和等于（$1 \times 0.1 + 2 \times 0.3 + 3 \times 0.6$），这表明较大的权重对应较大的通道依赖值。然而，如果输入时间序列的时间语义发生变化，例如交换 x_1 和 x_3 的值，这意味着较大的值被分配了较小的权重。

通过有效利用注意力机制，可以根据不同输入序列的时间依赖关系，动态调整通道独立（CI）与通道依赖（CD）的组合。将公式 4 的结果简化为 $Q = [w_{:1}X, w_{:2}X, w_{:3}X]$，并将另一种权重组合定义为 $K = [t_{:1}X, t_{:2}X, t_{:3}X]$，则利用 QK 发现时间上下文的过程可描述如下：

$$\begin{bmatrix} w_{:1}X & w_{:2}X & w_{:3}X \end{bmatrix} \begin{bmatrix} t_{:1}X \\ t_{:2}X \\ t_{:3}X \end{bmatrix} = \begin{bmatrix} t_{:1}w_{:1}X^2 + t_{:2}w_{:2}X^2 + t_{:3}w_{:3}X^2 \end{bmatrix} \tag{5}$$

仔细考察 QK 的组合后可以发现， X^2 的存在会使系数根据输入序列的时间语义而变化。引入平方运算并调整双重权重，能够自动调整对语义信息的注意力，使其具有不同的权重。从输入 X 中获得且具有特殊

语义含义的系数定义为 $context$。使用固定权重和 TSA权重进行聚合的区别如下:

$$context \cdot \left[x_1\ x_2\ x_3\right] W \quad \leftrightarrow \quad \left[x_1\ x_2\ x_3\right] W \tag{6}$$

面向 CI 和通道依赖的注意力调制。我们采用高效注意力 [31]对序列进行建模，因为其具有线性时间复杂度且效果良好，如下所示:

$$E(Q,K,V) = (KV^\top)^\top Q \tag{7}$$

$$\text{where}, K = \rho_k(X), V = \rho_v(X) \text{ and } Q = \rho_q(X) \tag{8}$$

$$\text{and}, K \in \mathbb{R}^{d_k \times d}, V \in \mathbb{R}^{d_v \times d} \text{and } Q \in \mathbb{R}^{d_q \times d}. \tag{9}$$

选择 $\rho_i(\cdot)$有两种方案。如果采用卷积映射，则 Q、 K和 V表示对 X建模得到的序列。另一方面，如果采用线性映射，则 Q、 K和 V表示单个 X序列的增强特征。这两种方法的直观解释如下:

$$W_{conv} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \quad \leftrightarrow \quad W_{linear} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \tag{10}$$

其中， $X = [x_1, x_2, x_3]$、 $Y = [y_1, y_2, y_3]$和 $Z = [z_1, z_2, z_3]$表示不同的序列，卷积权重 W_{conv} 与不同序列进行交互。线性权重 W_{linear} 则专注于增强单个序列的特征。然而，在实验中，我们没有观察到这两种方法之间存在显著差异。

3.3 面向通道独立与通道依赖的深度监督

基于注意力的机制通过融入输入特征的语义，能够调整模型对具有不同语义特征的输入的适应能力。然而，研究发现，当前该方法的最先进方案通常需要堆叠多个层才能提升性能，这会带来显著的计算开销。因此，我们提出了一种全新的优化视角，聚焦于模型的深层参数。通过微调这些参数，我们能够在降低计算成本的同时提升模型性能。这种方法使模型无需大量计算资源即可获得更好的性能。

参考图2左侧的 CI 和 CD 组件，其输出 X_{CI} 和 X_{CD} 均为 $\in \mathbb{R}^{S \times n}$。从抽象的角度来看，它们可以被视为两个独立的序列预测器，如图2所示。由于它们距离输出层较远，我们提出通过分别进行监督，来改善这两个独立序列预测器的深层参数拟合效果。

深层级上的 CI 和 CD 组件。具体而言，给定真实预测序列 y 以及两个独立预测器 X_{CI} 和 X_{CD} 的输出，该过程定义如下：

$$Distance_{\mathrm{CI}} = Loss(X_{\mathrm{CI}}, Y_{true}) \tag{11}$$
$$Distance_{\mathrm{CD}} = Loss(X_{\mathrm{CD}}, Y_{true}) \tag{12}$$

其中，$Distance_{\mathrm{CI}}$ 和 $Distance_{\mathrm{CD}}$ 分别表示 CI 和 CD 预测器的预测结果 X_{CI} 和 X_{CD} 与真实结果 Y_{true}之间的误差。我们的目标是最小化 $Distance_{\mathrm{CI}}$ 和 $Distance_{\mathrm{CD}}$。从更直观的角度来看，如果独立预测结果都非常准确，那么联合预测结果也很可能准确。

将最终预测目标定义为：

$$X_{CD+CI} = \mathrm{Process}(X_{CD}, X_{CI}) \tag{13}$$

其中，Process(·, ·)表示整合 CI 和 CD 后的处理过程。

因此，MLinear 的最终预测所生成的优化目标 $Distance_{\mathrm{CI+CD}}$ 为：

$$Distance_{\mathrm{CI+CD}} = Loss(X_{\mathrm{CI+CD}}, Y_{true}) \tag{14}$$

3.4 损失函数

虽然以往的模型都采用均方误差（MSE）作为损失函数，但我们的实验表明，Huber 损失和平均绝对误差（MAE）能够取得更好的表现。与 MSE 相比，Huber 损失对离群值具有更强的鲁棒性，因此能够更好地处理离群值并提升模型鲁棒性。与 MSE 相比，MAE 对离群值的惩罚相对较小，因此模型训练时收敛速度更快。

通过实证，我们提出了一种新颖的损失函数，结合了平均绝对误差和 Huber 损失的优点。值得注意的是，我们没有仔细调整融合权重，而是仅采用了简单的加法方法。尽管如此，在大多数测试中，它仍展现出显著的性能提升。不同损失函数之间的权衡将留待未来研究。我们提出的新损失函数如下所示：

$$L_\sigma(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 + |y - f(x)|, |y - f(x)| \le \sigma \\ (\sigma + 1)|y - f(x)| - \frac{1}{2}\sigma^2, \quad |y - f(x)| > \sigma \end{cases} \tag{15}$$

考虑到 CI 和 CD 的独立预测结果，Mlinear 的总损失函数可表示为：

$$Loss = L_\sigma(Y_{true}, f(X_{CD})) + L_\sigma(Y_{true}, f(X_{CI})) + L_\sigma(Y_{true}, f(X_{CI+CD})) \tag{16}$$

从目标函数可以看出，Mlinear 对隐藏层实施了深度监督，以生成更具区分性的特征。而在测试过程中，只会使用 X 来获得最终的预测结果。

3.5 模型分析

我们讨论了 Mlinear 与两种具有代表性的时间序列预测方法之间的差异——最新的 Transformer 方法 PatchTST，以及非 Transformer 方法 DLinear，旨在突出我们设计理念的独特之处。

与 PatchTST 的比较 PatchTST 是一种基于 Transformer 的时间序列预测方法。其主要设计理念是探索通道注意力在 Transformer 中的应用，并利用分块探索局部时间依赖关系，借鉴视觉 Transformer 中使用的局部语义概念。我们总结了我们的设计理念与 PatchTST 之间的差异。首先，在注意力机制的使用方面，PatchTST 旨在利用注意力机制对通道独立序列之间的局部时间依赖关系进行建模。相比之下，我们的注意力机制旨在针对通道信息（CI）和上下文信息（CD），联合建模各个输入时间序列的时间语义。其次，在通道独立（CI）和通道依赖（CD）的利用方面，PatchTST 的主要贡献是将通道独立引入 Transformer 模型。相比之下，我们充分利用通道独立和通道依赖的优势，并巧妙地将二者结合起来。

与 Dlinear 的比较 Dlinear 是一种非 Transformer 的简单线性组合时间序列预测方法。Dlinear 的主要思路是使用一个简单的线性层来预测一个通道独立（CI）的单个序列。相比之下，首先，与我们的方法和 PatchTST 之间的差异类似，我们的方法充分利用了 CI 和通道依赖（CD）的联合预测优势，而 Dlinear 不具备这一优势。其次，从模型设计的角度来看，Dlinear 在概念上较为简单，其性能主要来自对 CI 特性的充分利用。相比之下，我们巧妙地结合了 Transformer 和线性模型的优势：使用线性模型获得 CI 和 CD 的联合预测，并使用 Transformer 根据不同的时间语义自适应地调整 CI 和 CD 的权重。

时间复杂度分析 从图2可以观察到，mlinear 主要由两种不同类型的层组成，即线性层和注意力层。

考虑前 n 个线性层时，它们的全部权重为（L， S）。由于输入的维度（X_1，……， X_n）均为（L， v），单个线性层与输入的乘法复杂度为 O（vLS）。

对于第 $(n+1)$ 个线性层，由于输入 X_{n+1} 为 (L, v)，且第 $(n+1)$ 个线性个体的权重为 (L, S)，因此第 $(n+1)$ 个线性层的时间复杂度为 $O(LS)$。

对于第（$n+2$）个线性层，由于输入是 CI 和通道依赖的拼接，输入序列 X 的形状为（$2S$， v），而第（$n+2$）个线性层的权重形状为（$2S$， S）。因此，第（$n+2$）层的时间复杂度为 $O(2S^2)$。

考虑到高效注意力对 CI 和通道依赖的优势，我们引入高效注意力来计算 KVQ 阶，将基于序列输入长度的自注意力的二次时间复杂度降低至 $O(d_k d_v)$，从而使时间复杂度降至线性级别。在本文的具体实现中，$d_k = \frac{S}{P}$、$d_q = d_k$、$d_v = 2$，其时间复杂度显著低于传统的基于Transformer的模型。

4 实验

4.1 实验设置

数据集 我们在七个数据集上进行了测试，包括近年来广泛用作基准的电力、天气和疾病数据集。除由 Informer [42], 收集的 ETT 数据集系列外，其他数据集均可在 Autoformer [35]中找到。

1. ETT（电力变压器温度） [42] 数据集由四个不同粒度的数据集组成，包括小时级数据（ETTh1、ETTh2）和 15 分钟级数据（ETTm1、ETTm2）。每个数据集涵盖 2016 年 7 月至 2018 年 7 月的两年周期，包含七项与电力变压器的油和负荷相关的特征。2. 天气数据集包含 21 项天气指标，例如气温和湿度。其数据记录于 2020 年德国，每 10 分钟采集一次。3. 电力数据集记录了 2012 年至 2014 年间 321 位用户的每小时用电量。4. ILI 衡量流感样疾病患者占患者总人数的比例，包含美国疾病控制与预防中心于 2002 年至 2021 年收集的每周数据。

设置 为确保比较公平，我们在每个数据集上使用与 PatchTST [28]相同的批量大小。所有模型均遵循相同的实验设置，预测长度 T $\in \{24, 36, 48, 60\}$ 适用于 ILI 数据集，T $\in \{96, 192, 336, 720\}$适用于其他数据集，与原始 PatchTST 论文中的设置一致。基线结果取自 PatchTST 论文，因为该论文设定了严格的标准，以避免低估基线。我们使用均方误差和平均绝对误差作为多变量时间序列预测的评价指标。对于 MLinear，按照 PatchTST 的做法，我们使用两种回看窗口长度：336 和 512，以进行严格比较；在定量比较中，模型名称标记为 MLinear（336 或 512）。除定量实验外，其他实验均使用 MLinear（336）。为便于命名，在其他实验中我们将模型简称为“MLinear”，不加“336”后缀。所使用的损失函数如下：“L1-Loss”表示平均绝对误差损失，“L2-Loss”表示均方误差损失，“H-Loss”表示 Huber 损失，“M-Loss”表示本文提出的损失函数。

基线 我们比较的基线方法涵盖了当前最出色、最新的时间序列预测方法。具体而言，我们重点关注 PatchTST [28] 和 DLinear [40], ，因为它们目前分别是最出色的基于 Transformer 和非基于 Transformer 的长序列时间序列预测方法。

1. PatchTST [28] 是一种基于Transformer的时间序列预测模型，它将时间序列数据划分为多个小块（分块），并采用通道独立（CI）输入编码。2. DLinear [40] 充分利用通道独立属性，只需通过单个线性层处理单通道时间序列，便能取得优于大多数基于Transformer的模型的预测性能。3. Autoformer [35]首次在每个神经网络模块之后应用季节-趋势分解，这是一种时间序列分析中的标准方法，可使原始数据更易于预测。4. FEDformer [43] 在 Autoformer 的分解方案之上提出了专家混合策略，将通过不同尺寸的移动平均核提取的趋势组件结合起来。5. Informer [42] 分析了时间序列预测中的注意力机制，并提出稀疏注意力机制以节省计算和时间开销；同时，通过引入特征蒸馏和单步时间序列预测，相较于 Transformer 提升了效率和预测准确率。6. Pyraformer [24] 是一种用于时间序列预测的基于Transformer的模型。它能够捕捉多分辨率特征和时间依赖关系，在捕捉长期模式的同时保持较低的时间和空间复杂度。7. LogTrans 的 [19]核心组件是因果卷积自注意力机制，它通过因果卷积生成查询和键，从而更好地将局部上下文融入注意力机制。

4.2 性能比较与分析

定量实验七个广泛使用的长序列预测数据集上的对比结果见表 2。与最新的基线方法 PatchTST 保持一致，为了公平比较，我们使用两个版本的 MLinear，分别命名为 MLinear（输入序列长度），其输入序列长度分别为 336 和 512。从表 2 可以看出，输入序列长度分别为 336 和 512 的两个版本 MLinear，相较于其他基线方法均取得了显著的预测优势。

具体而言，MLinear (512) 实现最优性能的次数最多，达到 29 次，排名第二的次数为 18 次，显著优于具有相同输入序列长度的 PatchTST (512)，后者仅分别达到 10 次和 6 次，甚至优于 MLinear (336)。当输入序列长度限制为 336 时，MLinear 仍然保持着相对于 PatchTST 的显著性能优势，实现最优性能和次优性能的次数达到 21 次。

表 2. 使用监督式 PatchTST 得到的多变量长期预测结果。对于 ILI 数据集，预测长度 T 为 $\in \{24, 36, 48, 60\}$；对于其他数据集，预测长度 T 为 $\in \{96, 192, 336, 720\}$。最佳结果以粗体表示，次佳结果带下划线。在计数行中，第一个数字表示取得最佳性能的次数，第二个数字表示取得次佳性能的次数。

模型		MLinear(336)		MLinear(512)		PatchTST(336) ICLR 2023		PatchTST(512) ICLR 2023		DLinear AAAI 2023		Fedformer ICML 2022		Autoformer NIPS 2021		Informer AAAI 2021		Pyraformer ICLR 2022		LogTrans NIPS 2019	
评价指标		均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
Weather	96	0.146	0.188	**0.143**	**0.186**	0.152	0.199	0.149	0.198	0.176	0.237	0.238	0.314	0.249	0.329	0.354	0.405	0.896	0.556	0.458	0.490
	192	0.188	0.231	**0.185**	**0.230**	0.197	0.243	0.194	0.241	0.220	0.282	0.275	0.329	0.325	0.370	0.419	0.434	0.622	0.624	0.658	0.589
	336	0.240	0.273	**0.236**	**0.271**	0.249	0.283	0.245	0.282	0.265	0.319	0.339	0.377	0.351	0.391	0.583	0.543	0.739	0.753	0.797	0.652
	720	0.314	**0.322**	**0.308**	0.325	0.320	0.335	0.314	0.334	0.323	0.362	0.389	0.409	0.415	0.426	0.916	0.705	1.004	0.934	0.869	0.675
Electricity	96	0.133	0.225	0.131	0.223	0.130	0.222	**0.129**	**0.222**	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449	0.258	0.357
	192	0.149	0.239	**0.146**	**0.237**	0.148	0.240	0.147	0.240	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417	0.386	0.443	0.266	0.368
	336	0.165	0.255	**0.162**	**0.253**	0.167	0.261	0.163	0.259	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422	0.378	0.443	0.280	0.380
	720	0.203	0.288	**0.199**	**0.286**	0.202	0.291	**0.197**	0.290	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427	0.376	0.445	0.283	0.376
ILI	24	1.564	0.762	1.564	0.762	1.522	0.814	**1.319**	**0.754**	2.215	1.081	2.624	1.095	2.906	1.182	4.637	1.449	1.420	2.012	4.480	1.444
	36	1.543	**0.769**	1.543	**0.769**	**1.430**	0.834	1.579	0.870	1.963	0.963	2.516	1.021	2.585	1.038	4.650	1.463	7.394	2.031	4.799	1.467
	48	1.712	0.823	1.712	0.823	1.673	0.854	**1.553**	**0.815**	2.130	1.024	2.505	1.041	3.024	1.145	5.004	1.542	7.551	2.057	4.800	1.468
	60	1.710	0.834	1.710	0.834	1.529	0.862	**1.470**	**0.788**	2.368	1.096	2.742	1.122	2.761	1.114	5.071	1.543	7.662	2.100	5.278	1.560
ETTh1	96	**0.359**	**0.385**	0.362	0.388	0.375	0.399	0.370	0.400	0.375	0.399	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612	0.878	0.740
	192	**0.400**	**0.412**	0.402	0.414	0.414	0.421	0.413	0.429	0.405	0.416	0.423	0.446	0.456	0.457	1.007	0.786	0.790	0.681	1.037	0.824
	336	**0.417**	**0.421**	**0.417**	0.423	0.431	0.436	0.422	0.440	0.439	0.443	0.444	0.462	0.486	0.487	1.038	0.784	0.891	0.738	1.238	0.932
	720	**0.429**	**0.448**	0.430	0.450	0.449	0.466	0.447	0.468	0.472	0.490	0.469	0.492	0.515	0.517	1.144	0.857	0.963	0.782	1.135	0.852
ETTh2	96	0.274	**0.331**	**0.271**	**0.331**	0.274	0.336	0.274	0.337	0.289	0.353	0.332	0.374	0.332	0.368	1.549	0.952	0.645	0.597	2.116	1.197
	192	0.334	0.372	**0.332**	**0.370**	0.339	0.379	0.341	0.382	0.383	0.418	0.407	0.446	0.426	0.434	3.792	1.542	0.788	0.683	4.315	1.635
	336	0.332	0.381	**0.329**	0.381	0.331	**0.380**	0.329	0.384	0.448	0.465	0.400	0.447	0.477	0.479	4.215	1.642	0.907	0.747	1.124	1.604
	720	0.377	**0.413**	**0.375**	**0.414**	0.379	0.422	0.379	0.422	0.605	0.551	0.412	0.469	0.453	0.490	3.656	1.619	0.963	0.783	3.188	1.540
ETTm1	96	**0.289**	**0.329**	0.291	0.333	**0.290**	0.342	0.293	0.346	0.299	0.343	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510	0.600	0.546
	192	**0.331**	**0.354**	0.333	0.356	0.332	0.369	0.333	0.370	0.335	0.365	0.365	0.415	0.514	0.495	0.725	0.619	0.557	0.537	0.837	0.700
	336	0.369	**0.376**	**0.364**	**0.377**	0.366	0.392	0.369	0.392	0.369	0.386	0.392	0.425	0.510	0.492	1.005	0.741	0.754	0.655	1.124	0.832
	720	0.428	**0.410**	0.422	0.409	0.420	0.424	**0.416**	0.420	0.425	0.421	0.446	0.458	0.527	0.493	1.133	0.845	0.908	0.724	1.153	0.820
ETTm2	96	0.159	**0.243**	**0.159**	**0.243**	0.165	0.255	0.166	0.256	0.167	0.260	0.180	0.271	0.205	0.293	0.355	0.462	0.435	0.507	0.768	0.642
	192	0.215	0.283	**0.214**	**0.282**	0.220	0.292	0.223	0.296	0.224	0.303	0.252	0.318	0.278	0.336	0.595	0.586	0.730	0.673	0.989	0.757
	336	0.272	0.320	**0.266**	**0.316**	0.278	0.329	0.274	0.329	0.281	0.342	0.324	0.364	0.343	0.379	1.270	0.871	1.201	0.845	1.334	0.872
	720	**0.351**	**0.371**	0.357	0.376	0.367	0.385	0.362	0.385	0.397	0.421	0.410	0.420	0.414	0.419	3.001	1.267	3.625	1.451	3.048	1.328
计数		21	21	29	18	3	12	10	6	0	1	0	0	0	0	0	0	0	0	0	0

分别为 次，其表现显著优于 PatchTST（336），后者分别仅为 3 次和 12 次。此外，值得注意的是，考虑到 CI 和 CD 的特性，DLinear、PatchTST 和 MLinear 相较于其他对比模型取得了显著的领先优势。例如，在预测长度最长为 720 的天气数据集上，DLinear、PatchTST 和 MLinear 之间的均方误差（MSE）性能差距不超过 0.02。然而，与其他表现最佳的 Fedformer 相比，性能差距接近 0.08（Fedformer 的 MSE 为 0.389，而 Mlinear（512）的 MSE 为 0.307）。

CI 与 CD 混合消融实验 我们使用 L2 损失和 M 损失在电力、ETTh1 和 ETTm1 数据集上训练四种方法，并比较它们的性能，以展示深度监督相较于 CI 和 CD 的优势，以及 CI 与 CD 的简单组合的效果。从表 3 可以看出，在三个数据集上，完整版本的 MLinear 分别有 12 次（使用 L2 损失）和 18 次（使用 M 损失）取得最佳性能，而 CI 和 CD 仅分别有 6 次（使用 L2 损失）和 3 次（使用 M 损失）取得最佳性能。CI 与 CD 的简单组合分别有 5 次（使用 L2 损失）和 10 次（使用 M 损失）取得最佳性能。这表明，简单

Loss		L2 损失				M 损失			
评价指标		均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差	均方误差 平均绝对误差
方法		CD	CI	S-ATT	D-ATT	CD	CI	S-ATT	D-ATT
Electricity	96	0.140 0.236	0.134 0.229	0.134 0.229	**0.133 0.228**	0.141 0.234	0.135 **0.225**	0.134 **0.225**	**0.133 0.225**
	192	0.154 0.247	**0.150** 0.243	**0.150 0.242**	**0.150** 0.245	0.154 0.246	0.150 0.239	**0.149 0.239**	**0.149 0.239**
	336	0.170 0.264	0.167 **0.260**	0.167 0.261	**0.166** 0.262	0.170 0.262	0.167 **0.255**	0.166 **0.255**	**0.165 0.255**
	720	0.209 0.296	0.206 **0.293**	0.208 0.296	**0.204** 0.295	0.209 0.294	0.206 0.288	0.205 **0.287 0.203**	0.288
ETTh1	96	0.370 0.395	0.378 0.400	0.371 0.395	**0.366 0.394**	0.364 0.387	0.372 0.392	0.367 0.389	**0.359 0.385**
	192	0.407 **0.415**	0.416 0.423	0.415 0.425	**0.403** 0.418	0.406 **0.410**	0.413 0.417	0.407 0.414	**0.400** 0.412
	336	0.424 **0.425**	0.429 0.432	0.431 0.436	**0.418** 0.429	0.429 0.422	0.433 0.428	0.428 0.425	**0.417 0.421**
	720	**0.432 0.449**	0.447 0.456	0.454 0.464	0.451 0.465	0.425 0.441	0.440 0.45	0.434 **0.447 0.429**	0.448
ETTm1	96	0.301 0.343	0.289 0.336	**0.288 0.335**	**0.288** 0.337	0.297 0.337	0.292 0.330	0.291 **0.329 0.289**	0.329
	192	0.338 0.365	0.331 **0.360**	0.331 **0.360 0.327**	0.361	0.335 0.359	0.335 0.355	0.334 **0.354**	**0.331 0.354**
	336	0.371 0.384	0.370 **0.382**	0.372 0.385	**0.364** 0.383	**0.369** 0.379	0.373 0.377	0.372 **0.376 0.369**	0.376
	720	**0.427 0.416**	0.429 **0.416**	0.432 0.419	0.428 0.417	**0.427** 0.412	0.431 **0.411**	0.430 **0.411**	0.428 **0.410**
计数		6	6	5	12	3	3	10	18

表 3. CI、CD、未采用深度监督的 CI 与 CD 聚合，以及采用 CI 与 CD 聚合并结合深度监督的 MLinear 之间的直观比较。“S-ATT”指未采用深度监督、使用单一损失函数的 MLinear，而“D-ATT”指在 MLinear 中采用深度监督。使用的损失函数为 M 损失。

		H 损失		M 损失		L1 损失		L2 损失	
评价指标		均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
ETTh1	96	0.363	0.390	**0.359**	0.385	**0.359**	**0.384**	0.366	0.394
	192	0.402	0.415	0.400	0.412	**0.399**	**0.410**	0.403	0.418
	336	**0.417**	0.424	**0.417**	**0.421**	0.419	**0.421**	0.418	0.429
	720	0.435	0.453	**0.429**	0.448	0.430	**0.446**	0.451	0.465
ETTh2	96	**0.274**	0.333	**0.274**	**0.331**	0.276	**0.331**	0.283	0.340
	192	0.336	0.375	**0.334**	**0.372**	0.335	**0.372**	0.343	0.385
	336	0.333	0.385	0.332	0.381	**0.329**	**0.378**	0.340	0.395
	720	0.382	0.419	**0.377**	**0.413**	0.379	**0.413**	0.400	0.438
ETTm1	96	0.289	0.334	0.289	**0.329**	0.292	**0.329**	**0.288**	0.337
	192	0.329	0.357	0.331	**0.354**	0.334	**0.354**	**0.327**	0.361
	336	0.366	0.379	0.369	**0.376**	0.372	**0.376**	**0.364**	0.383
	720	**0.422**	0.413	0.428	**0.410**	0.430	**0.410**	0.428	0.416
ETTm2	96	**0.159**	0.246	**0.159**	**0.243**	0.160	**0.243**	0.162	0.249
	192	**0.210**	**0.280**	0.215	0.283	0.219	0.282	0.219	0.291
	336	**0.261**	**0.315**	0.272	0.320	0.274	0.319	0.273	0.325
	720	0.357	0.376	**0.351**	0.371	0.352	**0.370**	0.369	0.386
数量		8	13	17	11	17	8	3	2

表4. Mlinear 使用四种不同损失函数在四个数据集上的性能结果。

评价指标 均方误差 平均绝对误差		MSE	MAE	均方误差 平均绝对误差	MSE	MAE
Loss		S-L1 损失	D-L1 损失	S-L2 损失	D-L2-L损失函数	
ETTh1	92	0.379	0.402 0.359(↓0.020)0.384(↓0.018)	0.374	0.399 0.366(↓0.008)0.394(↓0.005)	
	192	0.414	0.420 0.399(↓0.015) 0.410(↓0.010)	0.430	0.436 0.403(↓0.027) 0.418(↓0.018)	
	336	0.435	0.432 0.419(↓0.016) 0.421(↓0.011)	0.452	0.456 0.418(↓0.034) 0.429(↓0.027)	
	720	0.450	0.463 0.430(↓0.020) 0.447(↓0.016)	0.475	0.482 0.451(↓0.024) 0.465(↓0.017)	
ETTh2	92	0.282	0.333 0.276(↓0.006)0.331(↓0.002)	0.294	0.350 0.283(↓0.011)0.340(↓0.010)	
	192	0.359	0.381 0.335(↓0.024) 0.372(↓0.009)	0.378	0.402 0.343(↓0.035) 0.385(↓0.017)	
	336	0.351	0.387 0.329(↓0.022) 0.378(↓0.009)	0.353	0.406 0.340(↓0.013) 0.395(↓0.009)	
	720	0.375	0.412 0.379(↓0.004) 0.413(-0.001)	0.411	0.447 0.400(↓0.011) 0.438(↓0.009)	
ETTm1	96	0.301	0.346 0.292(↓0.009)0.329(↓0.017)	0.305	0.355 0.288(↓0.017)0.337(↓0.018)	
	192	0.342	0.367 0.334(↓0.008) 0.354(↓0.013)	0.339	0.372 0.327(↓0.012) 0.361(↓0.011)	
	336	0.382	0.389 0.372(↓0.010) 0.376(↓0.013)	0.382	0.397 0.364(↓0.018) 0.383(↓0.014)	
	720	0.444	0.424 0.430(↓0.014) 0.410(↓0.014)	0.434	0.430 0.428(↓0.006) 0.417(↓0.013)	
ETTm2	96	0.163	0.243 0.161(↓0.002)0.243(↓0.000)	0.167	0.255 0.162(↓0.005)0.250(↓0.005)	
	192	0.218	0.283 0.219(↓0.001) 0.283(↓0.000)	0.234	0.305 0.219(↓0.015) 0.291(↓0.014)	
	336	0.287	0.326 0.274(↓0.013) 0.319(↓0.007)	0.290	0.344 0.273(↓0.017) 0.326(↓0.018)	
	720	0.381	0.389 0.352(↓0.011) 0.370(↓0.019)	0.382	0.399 0.370(↓0.012) 0.386(↓0.013)	
Loss		S-M 损失	D-M 损失	S-H 损失	D-H 损失	
ETTh1	92	0.380	0.403 0.359(↓0.021)0.385(↓0.018)	0.385	0.408 0.363(↓0.022)0.390(↓0.018)	
	192	0.410	0.417 0.400(↓0.010) 0.412(↓0.005)	0.424	0.429 0.402(↓0.020) 0.415(↓0.014)	
	336	0.442	0.440 0.417(↓0.025) 0.421(↓0.019)	0.443	0.443 0.417(↓0.026) 0.424(↓0.019)	
	720	0.459	0.468 0.429(↓0.030) 0.448(↓0.020)	0.460	0.470 0.435(↓0.025) 0.453(↓0.017)	
ETTh2	92	0.280	0.333 0.274(↓0.006)0.331(↓0.002)	0.287	0.342 0.274(↓0.013)0.333(↓0.009)	
	192	0.364	0.385 0.334(↓0.030) 0.372(↓0.013)	0.366	0.396 0.336(↓0.030) 0.375(↓0.021)	
	336	0.346	0.389 0.332(↓0.014) 0.381(↓0.008)	0.348	0.393 0.332(↓0.016) 0.385(↓0.008)	
	720	0.381	0.420 0.377(↓0.004) 0.413(↓0.007)	0.383	0.426 0.382(↓0.001) 0.419(↓0.007)	
ETTm1	92	0.296	0.343 0.290(↓0.006)0.330(↓0.013)	0.303	0.352 0.289(↓0.014)0.334(↓0.018)	
	192	0.343	0.371 0.331(↓0.012) 0.354(↓0.017)	0.344	0.374 0.329(↓0.015) 0.357(↓0.017)	
	336	0.386	0.392 0.369(↓0.017) 0.376(↓0.016)	0.384	0.395 0.366(↓0.018) 0.379(↓0.016)	
	720	0.440	0.421 0.428(↓0.012) 0.410(↓0.011)	0.435	0.422 0.422(↓0.013) 0.413(↓0.009)	
ETTm2	92	0.158	0.243 0.159(↑0.001)0.243(–0.000)	0.165	0.250 0.160(↓0.005)0.246(↓0.004)	
	192	0.222	0.287 0.215(↓0.007) 0.283(↓0.004)	0.225	0.293 0.211(↓0.014) 0.281(↓0.012)	
	336	0.287	0.330 0.272(↓0.015) 0.320(↓0.010)	0.290	0.338 0.262(↓0.028) 0.316(↓0.022)	
	720	0.379	0.390 0.351(↓0.028) 0.371(↓0.019)	0.380	0.395 0.357(↓0.023) 0.377(↓0.018)	

表5. 深度监督的消融实验使用了四种不同的损失函数，前缀“S-”表示简单混合实验，仅使用一个最终目标损失；前缀“D-”表示深度监督实验，对通道独立、通道依赖和最终目标进行监督。

方法		3ATT		简单		1ATT	
评价指标		均方误差	平均绝对误差	均方误差	平均绝对误差	均方误差	平均绝对误差
ETTh1	92	**0.359**	**0.386**	0.360	**0.386**	0.359	**0.385**
	192	**0.399**	**0.411**	0.402	0.412	0.400	0.412
	336	0.418	0.422	0.422	0.423	**0.417**	**0.421**
	720	0.435	0.455	**0.429**	**0.445**	**0.429**	0.448
ETTh2	92	0.277	0.332	0.277	**0.331**	**0.274**	**0.331**
	192	**0.333**	**0.372**	0.335	**0.372**	0.334	**0.372**
	336	0.334	0.382	**0.331**	**0.381**	0.332	**0.381**
	720	**0.373**	**0.411**	0.377	0.414	0.377	0.413
ETTm1	92	0.291	0.332	0.291	**0.330**	**0.289**	**0.329**
	192	**0.331**	0.355	0.332	**0.354**	**0.331**	**0.354**
	336	0.370	0.377	**0.369**	**0.376**	**0.369**	**0.376**
	720	**0.425**	**0.409**	0.428	0.410	0.428	0.410
ETTm2	92	0.161	0.244	0.161	0.245	**0.159**	**0.243**
	192	0.219	**0.282**	0.219	0.285	**0.215**	0.283
	336	**0.271**	**0.320**	0.274	0.323	0.272	**0.320**
	720	0.359	0.374	0.367	0.381	**0.351**	**0.371**
计数		14		11		20	

表6. CI 和 CD 的三种不同聚合方法的比较包括“简单”，即使用线性层进行聚合；“1ATT”和“3ATT”，分别表示使用一层或三层注意力进行聚合。所使用的损失函数为 M 损失。

仅结合通道独立和通道依赖无法实现大于各部分之和的协同效应。

损失函数消融实验 我们比较了使用 L2 损失、L1 损失、H 损失和 M 损失的 MLinear 性能，以突出 M 损失的优越性能。

从表4可以看出，使用 M 损失在 ETTh1 上结合 L2 损失三次取得最佳性能，结合 L1 损失一次取得最佳性能。在 ETTh2 上，结合 L2 损失三次、结合 L1 损失三次取得最佳性能。在 ETTm1 上，结合 L1 损失四次取得最佳性能。在 ETTm2 上，结合 L2 损失两次、结合 L1 损失一次取得最佳性能。

深度监督消融实验我们比较深度监督在四种不同损失函数上的有效性，以充分验证其在不同损失函数中的通用性和可扩展性。

从表5可以看出，与简单的单损失函数方法相比，深度监督在大多数情况下都能提升整体性能。例外情况包括：在预测长度为 720 的 ETTh1 上使用平均绝对误差，以及在预测长度为 192 的 ETTm2 上使用平均绝对误差；此外，在预测长度为 92 的 ETTm2 上使用 M 损失时，也未观察到明显的性能提升。值得注意的是，深度监督不会引入

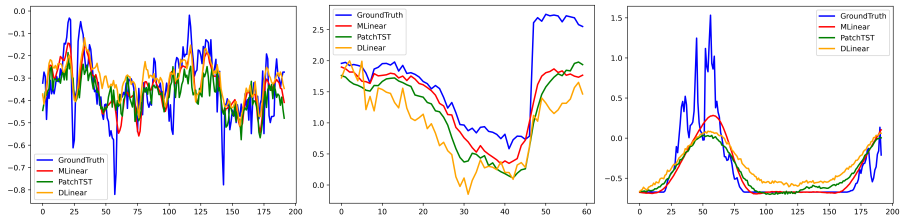


图 3. 可视化分析：从左到右依次展示电力变压器温度 ETTm1 数据集、ILI 数据集和天气数据集。这三个数据集的预测时域分别为 192、60 和 192。

任何额外参数，也不会影响推理效率，因此这种整体性能提升意义重大。

4.3 混合通道独立与通道依赖的不同融合方法

我们深入探究不同的聚合方法，并进行了三组对比，以实现两个研究目标：(1) 注意力聚合是否比线性聚合更有效？(2) 堆叠注意力层进行聚合能否提升预测性能？三项实验均使用完整版本的 MLinear，但将聚合方法分别替换为高效自注意力（ATT）、线性聚合和三层 ATT（3ATT）。

从表6可以看出，在 ETTh1、ETTh2、ETTm1 和 ETTm2 这四个数据集上，1ATT 有 19 次取得最佳性能，而简单聚合仅有 11 次取得最佳性能。值得注意的是，尽管增加了 ATT 层数，三层 ATT 的性能并不优于 1ATT，取得最佳性能的次数甚至更少。这表明，堆叠过多参数可能会降低 MLinear 在信息密度较高的长序列时间序列预测数据集上的泛化能力。

4.4 回看窗口分析

直观而言，分析更长的历史序列可以增加我们的预测知识。然而，先前的研究 PatchTST [28] 和 DLinear [40] 表明，在处理更长的输入时，基于 Transformer的模型性能会显著下降。为探究这一点，我们在三个数据集上使用均方误差和平均绝对误差作为评价指标进行了实验，并改变回看窗口，同时将预测跨度固定为 96 个时间步。结果如图4所示。图4揭示了两个趋势：（1）随着输入序列长度增加到 336 和 720，Transformer、Informer、Fedformer 和 Autoformer 的性能均显著下降。（2）DLinear、PatchTST 和 MLinear （我们提出的模型）随着输入

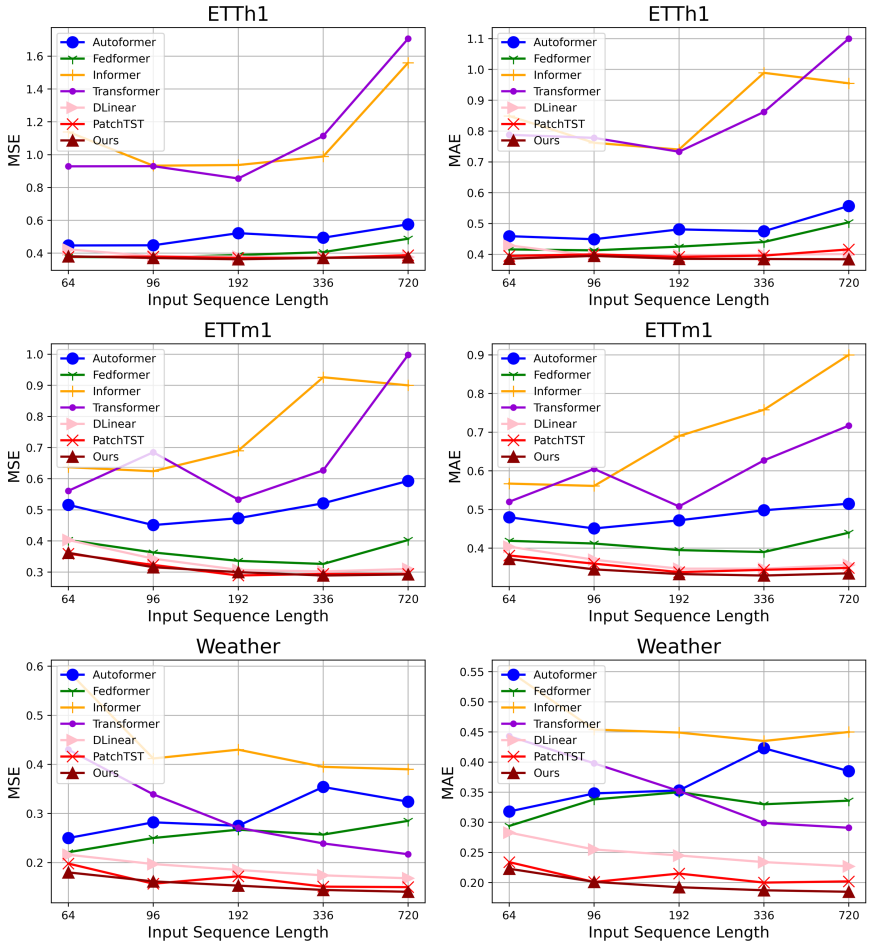


图 4. Weather 数据集上不同回看窗口的预测性能（均方误差和平均绝对误差）。回看窗口分别设为 64、96、192、336、720，预测跨度为 96。

序列长度增加，表明这些模型可以从更长的输入序列中获益，提升知识容量。此外，MLinear 的绝对性能相对优于 PatchTST 和 DLinear。

4.5 效率比较

我们对不同模型进行了效率比较。为了使比较结果更加明显，所有测试均使用各模型作者发布的代码，在 RTX 3070 GPU 上进行；ETTh1 的输入序列长度为 336，预测长度为 96。

表7显示，与其他基线方法相比，DLinear 和 MLinear（我们提出的模型）均实现了显著的效率提升。在参数数量方面，MLinear 的参数量少于 DLinear。然而，为调整 CI 和 CD 而引入注意力机制增加了训练和推理时间的计算开销，因此与 DLinear 相比，训练和推理时间没有显著差异。

表 7。效率比较（每个训练轮次）。

	参数	训练时间	推理时间
Informer	11.32M	22.97s	2.87s
Autoformer	10.53M	32.84s	7.27s
Transformer	10.54M	26.83s	3.13秒
PatchTST	20.78M	21.76秒	2.57秒
DLinear（单变量）	0.452M	1.03秒	0.29秒
MLinear（我们的方法）	0.277M	1.14秒	0.29秒

5 结论与未来工作

本文提出了 MLinear，并探讨了一个尚未解决的问题：我们能否有效利用通道独立性（CI）和通道依赖性（CD），提升长序列时间序列预测（LSTF）的性能？MLinear 的两项设计原则对此给出了明确答案：（1）根据输入语义进行时间语义感知调优，调整 CI 和 CD；（2）将 CI 和 CD 视为独立的预测器，并对其应用深度监督。此外，我们通过实验为时间序列预测任务提出了一种有效的损失函数。在七个广泛使用的数据集上的实验结果表明，与最先进的基线方法相比，我们提出的 MLinear 在预测性能上具有显著优势。此外，MLinear 的效率也处于顶尖水平，在参数使用量和推理时间两方面均显著优于当前基于 Transformer 的方法。

参考文献

1. Ariyo, A.A., Adewumi, A.O., Ayo, C.K.：使用 ARIMA 模型预测股票价格。载于：2014 年 UKSim-AMSS 第 16 届国际计算建模与仿真会议。第 106–112 页。IEEE（2014） 2. Bai, S., Kolter, J.Z., Koltun, V.：用于序列建模的通用卷积网络和循环网络的实证评估。arXiv 预印本 arXiv:1803.01271（2018） 3. Cai, D., Lam, W.：用于图到序列学习的图变换器。载于：AAAI人工智能大会论文集。第 34 卷，第 7464–7471 页（2020） 4. Cai, L., Janowicz, K., Mai, G., Yan, B., Zhu, R.：交通变换器：捕捉时间序列的连续性和周期性以进行交通流预测。地理信息系统汇刊**24**(3), 736–755（2020） 5. Cirstea, R.G., Guo, C., Yang, B., Kieu, T., Dong, X., Pan, S.：Triformer：用于长序列多变量时间序列预测的三角形变量特定注意力——完整版。arXiv 预印本 arXiv:2204.13767（2022） 6. Ding, Q., Wu, S., Sun, H., Guo, J., Guo, J.：用于股票走势预测的分层多尺度高斯变换器。载于：IJCAI。第 4640–4646 页（2020） 7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., 等：一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer。arXiv 预印本 arXiv:2010.11929（2020） 8. Dudukcu, H.V., Taskiran, M., Taskiran, Z.G.C., Yildirim, T.：结合循环神经网络方法的时间卷积网络，用于混沌时间序列预测。《应用软计算》 **133**, 109945（2023） 9. Fan, J., Zhang, K., Huang, Y., Zhu, Y., Chen, B.：用于多变量时间序列预测的并行时空注意力时间卷积网络。《神经计算与应用》第 1–10 页（2021） 10. Friedman, J.H.：贪心函数逼近：梯度提升机。《统计年鉴》第 1189–1232 页（2001） 11. Geng, X., He, X., Xu, L., Yu, J.：用于多变量时间序列预测的图相关注意力循环神经网络。《信息科学》**606**, 126–142 （ 2022 ） 12. Guo, M.H., Xu, T.X., Liu, J.J., Liu, Z.N., Jiang, P.T., Mu, T.J., Zhang, S.H., Martin, R.R., Cheng, M.M., Hu, S.M.：计算机视觉中的注意力机制：综述。《计算视觉媒体》 **8**(3), 331–368（2022） 13. Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., Liu, Y.：利用本地气象站时间序列数据进行有效天气预报的时间卷积神经网络（TCN）。《软计算》 **24**, 16453–16482（2020） 14. Hewamalage, H., Bergmeir, C., Bandara, K.：用于时间序列预测的循环神经网络：现状与未来方向。《国际预测学杂志》 **37**(1), 388–427（2021） 15. Hochreiter, S., Schmidhuber, J.：长短期记忆网络。《神经计算》**9**(8), 1735–1780（1997） 16. Khaldi, R., El Afia, A., Chiheb, R., Tabik, S.：哪种循环神经网络单元结构最适合预测各类时间序列的行为？《应用专家系统》**215**, 119140（2023） 17. Lei, S., Shi, Z., Mo, W.：用于遥感图像超分辨率的基于 Transformer 的多阶段增强。《IEEE地球科学与遥感汇刊》 **60**, 1–11（2021）

18. Li, Q., Chen, Y., Zeng, Y.：用于遥感图像目标检测的迁移卷积神经网络 Transformer。Remote Sensing **14**(4), 984 (2022) 19. Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X., Yan, X.：增强时间序列预测中 Transformer 的局部性并打破其内存瓶颈。神经信息处理系统进展 **32** (2019) 20. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.：使用 Swin Transformer 进行图像复原。载于：IEEE/CVF国际计算机视觉会议论文集。第1833–1844页 (2021) 21. Lin, L., Fan, H., Zhang, Z., Xu, Y., Ling, H.：SwinTrack：一种简单而强大的 Transformer 跟踪基线。神经信息处理系统进展**35**, 16743–16754 (2022) 22. Lin, T., Wang, Y., Liu, X., Qiu, X.：Transformer 综述。AI Open (2022) 23. Liu, M., Zeng, A., Xu, Z., Lai, Q., Xu, Q.：时间序列是一种特殊序列：基于样本卷积与交互的预测。arXiv预印本 arXiv:2106.093051(9) (2021) 24. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.：Pyraformer：用于长程时间序列建模与预测的低复杂度金字塔注意力。载于：国际学习表征会议 (2021) 25. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., 等：Swin Transformer V2：扩展容量与分辨率。载于：IEEE/CVF计算机视觉与模式识别会议论文集。第12009–12019页 (2022) 26. Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., Zhao, P., Huang, J., Ananiadou, S., Rong, Y.：图变换器：从架构视角进行综述。arXiv预印本 arXiv:2202.08455 (2022) 27. Ni, H., Wang, S., Cheng, P.：一种基于推文嵌入和历史价格的股票趋势预测混合方法。万维网 **24**, 849–868 (2021) 28. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.：一个时间序列相当于64个词：使用 Transformer 进行长期预测。arXiv预印本 arXiv:2211.14730 (2022) 29. Petneh´azi, G.：用于时间序列预测的循环神经网络。arXiv预印本 arXiv:1901.00069 (2019) 30. Shamshad, F., Khan, S., Zamir, S.W., Khan, M.H., Hayat, M., Khan, F.S., Fu, H.：医学影像中的 Transformer：综述。Medical Image Analysis 第102802页 (2023) 31. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.：高效注意力：具有线性复杂度的注意力。载于：IEEE/CVF冬季计算机视觉应用会议论文集。第3531–3539页 (2021) 32. Valanarasu, J.M.J., Oza, P., Hacihaliloglu, I., Patel, V.M.：Medical Transformer：用于医学图像分割的门控轴向注意力。载于：医学图像计算与计算机辅助干预——MICCAI 2021：第24届国际会议，法国斯特拉斯堡，2021年9月27日至10月1日，论文集，第一部分24。第36–46页。Springer (2021) 33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.：注意力机制就是一切。神经信息处理系统进展 **30** (2017) 34. Wang, Y., Chen, J., Chen, X., Zeng, X., Kong, Y., Sun, S., Guo, Y., Liu, Y.：基于TCN-LightGBM的工业客户短期负荷预测。IEEE电力系统汇刊 **36**(3), 1984–1997 (2020)

35. Wu, H., Xu, J., Wang, J., Long, M.：Autoformer：结合自相关机制进行长期序列预测的分解式 Transformer。神经信息处理系统进展 **34**, 22419–22430 (2021) 36. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.：用于交通流量预测的时空 Transformer 网络。arXiv预印本 arXiv:2001.02908 (2020) 37. Xu, X., Feng, Z., Cao, C., Li, M., Wu, J., Wu, Z., Shang, Y., Ye, S.：一种改进的、基于 Swin Transformer 的遥感目标检测与实例分割模型。Remote Sensing **13**(23), 4779 (2021) 38. Yan, H., Ma, X., Pu, Z.：利用 Transformer 学习动态层次化交通时空特征。IEEE智能交通系统汇刊**23**(11), 22386–22399 (2021) 39. Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J.：图 Transformer 网络。神经信息处理系统进展 **32** (2019) 40. Zeng, A., Chen, M., Zhang, L., Xu, Q.：Transformer 对时间序列预测是否有效？arXiv预印本 arXiv:2205.13504 (2022) 41. Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P.：用于股票走势预测的基于 Transformer 的注意力网络。《应用专家系统》**202**, 117239 (2022) 42. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.：Informer：超越高效 Transformer 的长序列时间序列预测。载于：AAAI人工智能大会论文集。第35卷，第11106–11115页 (2021) 43. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.：Fedformer：用于长期序列预测的频率增强分解式 Transformer。载于：国际机器学习会议。第27268–27286页。PMLR (2022)