采用非线性评分与符号回归实现节能信道编码的KAN自编码器

AnthonyJoseph Perre、Parker Huggins 和 Alphan Şahin，南卡罗来纳大学电气工程系，美国南卡罗来纳州哥伦比亚。电子邮箱：{aperre, parkerkh}@email.sc.edu，asahin@mailbox.sc.edu

arXiv:2601.01598v1 [eess.SP] 4 Jan 2026

摘要——本文研究了将符号回归（SR）用于节能信道编码的 Kolmogorov-Arnold网络自编码器（KAN-AE）。通过使用符号回归，我们将 KAN自编码器转换为符号表达式，从而实现低复杂度实现，并提高无线电设备的能效。为进一步提升能效，我们在 SR过程中引入了一种新的非线性评分项，以便在可能的情况下帮助选择复杂度更低的方程。通过数值仿真，我们证明，结合 SR 后，KAN自编码器在提升能效的同时，能够实现有竞争力的块错误率性能。我们使用所提出的非线性度量对 KAN自编码器实现的能效进行评分，并将其与多层感知机自编码器（MLP-AE）进行比较。实验结果表明，结合 SR 的 KAN自编码器能耗比 MLP-AE 低 1.38 倍，这说明 KAN自编码器是实现节能型基于深度学习的信道编码的一种有前景的选择。

索引词—自编码器、信道编码、能效、Kolmogorov-Arnold 网络

I. 引言

深度学习（DL）已被证明能够增强现代无线通信系统中的传统信号处理技术。例如，DL 方法分别用于提升 [1] 中的信道估计，以及识别 [2], 中的调制类型。此外，DL 已应用于端到端正交频分复用（OFDM）系统中的联合信源信道编码任务学习 [1, 2]。尽管深度学习模型在无线通信领域的应用前景广阔，但在实际系统中的采用仍面临诸多挑战。主要问题之一在于移动设备硬件：受限的内存资源和 CPU能力会降低无线电设备上大型深度学习模型的效能[3]。包含数十万个可学习参数的复杂模型会给移动设备带来内存和时序问题，进而导致能量与功耗增加 [4]。因此，亟需更高效的模型架构，在不牺牲性能的前提下降低内存使用量和计算负载。

多层感知机（MLP）是各种深度学习架构的基本组成部分。最近，一种名为 Kolmogorov-Arnold 网络（KAN）的新型深度学习结构作为多层感知机的替代方案应运而生 [5]。研究表明，与多层感知机相比，Kolmogorov-Arnold 网络在准确率方面表现更优，同时所需的总参数量更少 [5]。 [6] 的作者对 [5]，中提出的一些论断进行了质疑，但确认在公平比较下，Kolmogorov-Arnold 网络在符号公式表示方面优于多层感知机。近年来，Kolmogorov-Arnold 网络已广泛应用于物理学、时间序列预测等多个领域，尤其得益于其更强的可解释性和符号表示能力 [7, 8]。本研究尤其关注 Kolmogorov-Arnold 网络所需参数数量较少这一特点，这使其更适合应用于移动设备等资源受限环境，因为这些设备的内存和计算资源有限。此外，Kolmogorov-Arnold 网络与符号回归高度兼容，这意味着所学习的网络可以表示为多个更简单的符号方程的组合。这些表达式能够通过降低模型执行的复杂度和资源需求来提升能效。

本研究探讨了自编码器（AEs）在信道编码中的应用，已有若干研究对此进行了讨论 [9, 10, 11]。在我们的方法中，我们将自编码器结构中的多层感知机替换为 Kolmogorov-Arnold 网络，并发现这种做法在总参数量更少的同时，块错误率（BLER）表现相当。整个 KAN自编码器训练完成后，我们使用符号回归推导表示其学习行为的方程。此外，我们在符号回归过程中引入非线性评分项，以便在可行时促使方程更加简洁。我们采用符号回归并结合所提出的非线性评分项，旨在降低自编码器模型的能耗。通过使用 Kolmogorov-Arnold 网络，我们表明，在不牺牲性能的情况下，某些深度学习架构可以降低能耗；这说明，对于无线通信中的特定深度学习任务，Kolmogorov-Arnold 网络可以成为多层感知机的一种有用替代方案。

论文结构安排如下。第二节介绍系统模型，并阐述与 Kolmogorov-Arnold 网络相关的基本概念。第三节介绍所提出的 KAN自编码器以及用于评估能效的度量。第四节展示块错误率性能，并比较各模型的能效。第五节总结全文。

记号：实数集和复数集分别记为

本研究得到美国国家科学基金会（NSF）通过资助项目 CNS-2438837 的支持。

$\mathbb{R}$ 和 $\mathbb{C}$。 $z = a + jb$ 的复共轭表示为 $z^* = a - jb$。 均值为零、方差为 σ^2 的圆对称复正态分布表示为 $\mathcal{CN}\left(0, \sigma^2\right)$。矩阵 $\mathbf{A}$ 的厄米共轭记为 $\mathbf{A}^{\mathrm{H}}$。 $\mathbb{E}[X]$表示 X 的期望值。

II. 系统模型

本节介绍 Kolmogorov-Arnold 网络的相关预备知识，并给出基于 OFDM 的自编码器的系统模型。

A. Kolmogorov-Arnold 网络

Kolmogorov-Arnold 网络的结构受到 Kolmogorov-Arnold 表示定理的启发。该定理指出，任意多元连续函数都可以表示为若干一元连续函数之和 [12], 即,

$$f(x_1, x_2, \ldots, x_d) = \sum_{i=0}^{2d} \Phi_i \left(\sum_{j=1}^{d} \phi_{i,j}(x_j) \right) , \tag{1}$$

其中 $\phi_{i,\ j}\colon [0,1] \to \mathbb{R}$ 且 $\Phi_i\colon \mathbb{R} \to \mathbb{R}$。 [5]中的作者将式 (1) 的内外求和推广为可容纳任意数量的层 L ，形式如下:

$$\begin{aligned} \mathrm{KAN}(\mathbf{x}) &= (\mathbf{\Phi}^{(\mathrm{L})} \circ \mathbf{\Phi}^{(\mathrm{L}-1)} \circ \cdots \circ \mathbf{\Phi}^{(2)} \circ \mathbf{\Phi}^{(1)})(\mathbf{x}) \\ &= \sum_{i_{L-1}=1}^{d_{L-1}} \phi_{i_L, i_{L-1}}^{(L)} \left(\cdots \sum_{i_1=1}^{d_1} \phi_{i_2, i_1}^{(2)} \left(\sum_{i_0=1}^{d_0} \phi_{i_1, i_0}^{(1)} \left(x_{i_0} \right) \right) \cdots \right) , \end{aligned} \tag{2}$$

其中 $\phi_{i_l}^{(l)}{}_{,\ i_{l-1}}(\cdot)$ 是第 l 层中连接第 i_{l-1} 个输入神经元与第 i_l 个输出神经元的可学习函数，即一条边； d_l 是第 l 层中的神经元数量， $\mathbf{\Phi}^{(l)}$ 包含第 l 层的所有可学习激活函数，形式如下:

$$\mathbf{\Phi}^{(l)}(\cdot) \triangleq \begin{bmatrix} \phi_{1,1}^{(l)}(\cdot) & \cdots & \phi_{1,d_{l-1}}^{(l)}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{d_l,1}^{(l)}(\cdot) & \cdots & \phi_{d_l,d_{l-1}}^{(l)}(\cdot) \end{bmatrix}, \tag{3}$$

并作用于前一层的输出$\mathbf{a}^{(l-1)} \in \mathbb{R}^{d_{l-1}}$，对于 $\forall l \in \{1, \ldots, L\}$, 有

$$\mathbf{a}^{(l)} = \mathbf{\Phi}^{(l)}(\mathbf{a}^{(l-1)}) \triangleq \begin{bmatrix} \sum_{i_{l-1}=1}^{d_{l-1}} \phi_{i_l, i_{l-1}}^{(l)} \left(a_{i_{l-1}}^{(l-1)} \right) \\ \vdots \\ \sum_{i_{l-1}=1}^{d_{l-1}} \phi_{i_l, i_{l-1}}^{(l)} \left(a_{i_{l-1}}^{(l-1)} \right) \end{bmatrix} .$$

这里，边是指输入神经元与输出神经元之间的连接，它会对输入执行某种变换。在多层感知机中，边上的权重是可学习的，而激活函数是固定的；Kolmogorov-Arnold 网络则恰好相反。

在 [5], 中，作者将激活函数 $\phi(x)$ 表示为 B样条和 Sigmoid 线性单元（SiLU）的线性组合, 即,

$$\phi(x) = w_{\mathrm{b}} \times \mathtt{SiLU}(x) + w_{\mathrm{s}} \times \sum_i \left(c_i \times B_i(x) \right) , \tag{4}$$

其中， $B_i(x)$是B样条基函数，由 p 次分段多项式构成，并由可学习权重 c_i进行缩放。参数 w_{b} 和 w_{s} 也可学习。每个B样条都定义在特定的网格区间上，该区间由输入样本的值域决定。

B. 系统模型

考虑一条单用户通信链路。设 $r = k/n$为该通信链路的速率，其中 k 为每条消息的信息比特总数， n 为信道使用次数总数。设$\mathbf{s}_{\mathrm{m}} \in \mathbb{R}^{2^k}$ 为消息 m的独热编码（ OHE）向量表示。编码器网络 $\epsilon(\mathbf{s}_{\mathrm{m}})$将$\mathbf{s}_{\mathrm{m}}$ 映射为向量$\mathbf{s}_{\mathrm{e}} \in \mathbb{R}^n$，随后将其转换为 $s_{\mathrm{tx}} \in \mathbb{C}^{n/2}$的实部和虚部，其中 $\mathbb{E}[|s_{\mathrm{tx}}|^2] = 1$。我们考虑编码器中包含 L_{enc} 层。对于基于多层感知机的神经网络，有

$$\mathbf{a}_{\mathrm{enc}}^{(l)} = \sigma_{\mathrm{enc}}^{(l)}(\mathbf{W}_{\mathrm{enc}}^{(l)} \mathbf{a}_{\mathrm{enc}}^{(l-1)} + \mathbf{b}_{\mathrm{enc}}^{(l)}) , \quad l = 1, \ldots, L_{\mathrm{enc}} , \tag{5}$$

其中， $\sigma_{\mathrm{enc}}^{(l)}$ 是逐元素非线性激活函数，$\mathbf{W}_{\mathrm{enc}}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ 和$\mathbf{b}_{\mathrm{enc}}^{(l)} \in \mathbb{R}^{d_l}$ 分别是权重矩阵和偏置向量。注意，对于多层感知机和KAN模型，$\mathbf{a}_{\mathrm{enc}}^{(0)} = \mathbf{s}_{\mathrm{m}}$ 和$\mathbf{a}^{(L_{\mathrm{enc}})}{}_{\mathrm{enc}} = \mathbf{s}_{\mathrm{e}}$。然而，对于基于KAN的神经网络，将式（2）重新表示后可得

$$\mathbf{a}_{\mathrm{enc}}^{(l)} = \mathbf{\Phi}_{\mathrm{enc}}^{(l)}(\mathbf{a}_{\mathrm{enc}}^{(l-1)}) , \quad l = 1, \ldots, L_{\mathrm{enc}} , \tag{6}$$

其中， $\mathbf{\Phi}_{\mathrm{enc}}^{(l)}$ 是 $d_l \times d_{l-1}$ 函数空间，且$\mathbf{a}_{\mathrm{enc}}^{(0)} = \mathbf{s}_{\mathrm{m}}$。本研究使用式（4）学习Kolmogorov-Arnold 网络中的激活函数。

发射机编码后，发射符号 s_{tx} 通过通信信道传播，并受到信道以及零均值圆对称复加性白高斯噪声（AWGN）的干扰。令 h 表示信道系数。接收符号 s_{rx} 可表示为

$$s_{\mathrm{rx}} = h s_{\mathrm{tx}} + w , \tag{7}$$

对于 $w \sim \mathcal{CN}\left(0, \sigma_n^2\right)$。在AWGN信道下，令$h = 1$。对于平坦衰落瑞利信道（例如OFDM子载波所经历的信道），我们则将 $h \sim \mathcal{CN}(0,1)$建模为，并假设接收机已知 h 。随后，使用最小均方误差（MMSE）均衡器对 s_{rx} 进行均衡，得到$\hat{s}_{\mathrm{rx}} = \frac{h^* s_{\mathrm{rx}}}{|h|^2 + \sigma_n^2}$，其中$\hat{s}_{\mathrm{rx}}$ 是均衡后的估计发射符号。然后，将$\hat{s}_{\mathrm{rx}} \in \mathbb{C}^{n/2}$ 的实部和虚部转换为单个实值向量$\mathbf{s}_{\mathrm{d}} \in \mathbb{R}^n$。

令 $\delta(\mathrm{s_d})$ 表示将 $\mathrm{s_d}$ 映射为对数优势向量 $\hat{}\mathbf{s}_{\mathrm{m}} \in \mathbb{R}^{2^k}$ 的解码器网络。对于 L_{dec} 层，基于多层感知机和基于 KAN 的解码器分别遵循如下结构

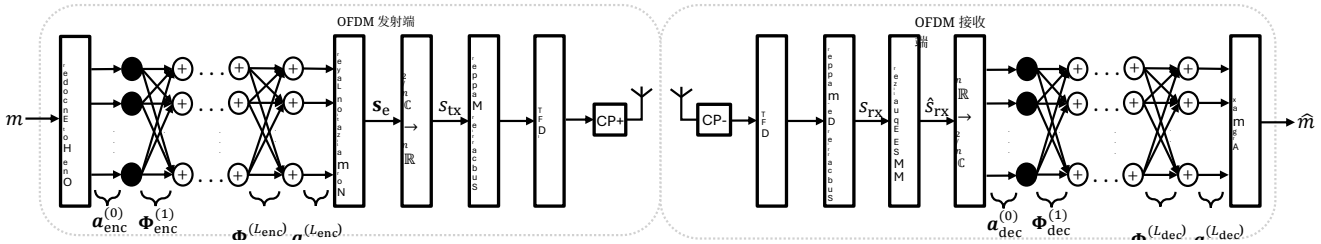


图1. 采用（n, k）KAN自编码器的正交频分复用发射机和接收机框图。

分别见式 (5) 和 (6)。此外，我们定义 $\mathbf{a}^{(0)}$ 解码器 $= \mathbf{s}_{\mathrm{d}}$ 和 $\mathbf{a}^{(L_{\mathrm{dec}})}$ 解码器 $= \hat{\mathbf{s}}_{\mathrm{m}}$。检测到的消息 $\hat{m}$ 表示为

$$\hat{m} = \arg \max_{\hat{\mathbf{s}}_{\mathrm{m}} \in \mathbb{R}^{2^k}} \hat{\mathbf{s}}_{\mathrm{m}} \ . \tag{8}$$

传统的单层神经网络也称为感知机，只能学习线性决策边界，因此难以捕捉复杂的非线性关系。多层感知机通过堆叠多个层来解决这一局限，而单个KAN层凭借其可学习激活函数的灵活性，便能够对复杂的非线性模式进行建模。图1展示了给定 n 次信道使用和 k 比特时，端到端正交频分复用发射机和接收机中的KAN自编码器。尽管我们考虑发射机和接收机中采用任意数量的KAN层，但在两种情况下都可以使用单层。

III. 节能型KAN自编码器

在本研究中，我们旨在减少编码器和解码器中的可学习参数数量，以提升发射机和接收机的能效，同时保持较低的块错误率。我们利用Kolmogorov-Arnold网络与符号回归的高度兼容性，在符号回归过程中引入新的惩罚项，以抑制能效较低的符号表达式。我们通过衡量非线性程度对函数进行启发式评分，详见第三节A和第三节B。我们的方法能够对Kolmogorov-Arnold网络和多层感知机网络的能效进行评分，不受具体实现方式的影响，详见第三节C。

A. 函数非线性度量

为了量化函数$f(x)$在区间 $[a,b]$上的非线性程度，我们提出采用分段线性近似。其基本思想是，根据在指定误差容限 ϵ内近似 $f(x)$所需的最少线性线段数 N，来评估 $f(x)$的非线性。线段数 N随后作为非线性的度量。 N 越大，表示非线性越高；而 N 越小，则意味着 $f(x)$在 $[a,b]$上越接近线性形式。

为表示上述度量，考虑一组均匀分布的分割点 a_1, a_2, ... a_{N+1}，其中 $a_1 = a$ 和 $a_{N+1} = b$，且 $[a_j, a_{j+1})$ 是第 j 个子区间在该子区间上， $f(x)$ 采用线性近似。第 j 个子区间 $[a_j, a_{j+1})$ 上的近似误差表示为

$$e_j = \int_{a_j}^{a_{j+1}} |f(x) - \psi_j(x)|^2 \, dx \ , \tag{9}$$

其中， $\psi_j(x) = m_j x + k_j$ 是 $f(x)$在 $[a_j, a_{j+1}$上的最佳拟合线性近似。为获得 $\psi_j(x)$，我们在第 j个子区间内对 $f(x)$进行过采样，并应用最小二乘回归，其中 m_j 和 k_j 分别为最佳拟合斜率和截距。随后，我们测量所有子区间上的近似误差，得到

$$\mathrm{E}(N) = \sum_{j=1}^{N} e_j \ . \tag{10}$$

$f(x)$ 的非线性度量记为 $Q[f(x)],$，它是满足 $\mathrm{E}(N) < \epsilon$ 的最小 N，即:

$$Q[f(x)] = \arg\min_{N} \ \mathrm{E}(N) \quad \text{s.t.} \quad \mathrm{E}(N) < \epsilon \ . \tag{11}$$

如果 $f(x)$ 表现出更强的非线性，则需要更大的 N 才能达到相同的近似精度。相反，如果 $f(x)$ 更具线性，则所需的 N 更小。 $Q[f(x)]$ 的度量是在符号回归的背景下建立的。非线性函数通常计算密集且能耗较高。因此，通过确定 $Q[f(x)],$，我们可以估计近似 $f(x)$ 的能耗，并在可行时引导符号回归采用更简单的近似。

示例 1：设 $f(x) = |5x|$ 和 $g(x) = \sin(5x)$ 定义在区间 $[-1,1]$ 上，并假设误差容限为 $\epsilon = 10^{-3}$。对于 $f(x)$ 和 $g(x)$，使用公式 (9) 和 (10) 计算 $\mathrm{E}(N)$。重复此过程，并在每次迭代中增加 N，直到满足公式 (10) 中的条件。然后，使用公式 (11) 确定每个函数的得分，得到 $Q[f(x)] = 2$ 和 $Q[g(x)] = 11$。这是预期结果，因为与 $|5x|$ 相比，$\sin(5x)$ 在 $[-1,1]$ 上的振荡性强得多，因此其非线性也更强。

B. 非线性约束下的符号回归

设激活函数 $\phi(x) \in [a,b] \to \mathbb{R}$ 和有限个候选函数 $\{f_k(x)\}_{k=1}^{K}$（例如 sin、log、exp）。获取样本 $S(\phi) = \{\phi(x_i) \mid x_i \in [a,b]\}$。令 $\hat{\phi}(x) = \gamma_{\mathrm{o}} f_k(\gamma_{\mathrm{i}} x + \beta_{\mathrm{i}}) + \beta_{\mathrm{o}}$ 为 $\phi(x)$ 的近似

Algorithm 1 Convert $\phi(x)$ to symbolic expression

Input: $S(\phi)$, $\{f_k(x)\}_{k=1}^K$
Output: $\phi_{\text{sym}}(x)$
$R^2_{\text{best}} = -\infty$; $Z_{\text{best}} = -\infty$; $\hat{\phi}_k(x) = \text{None}$; $\phi_{\text{sym}}(x) = \text{None}$
for $f_k(x)$ in $\{f_k(x)\}_{k=1}^K$ do
  $R^2_k = -\infty$
  for $(\gamma_{\text{i}}, \beta_{\text{i}})$ in grid $[-10, 10]$ do
    Set $\tilde{\phi}(x) = \gamma_{\text{o}} f_k(\gamma_{\text{i}} x + \beta_{\text{i}}) + \beta_{\text{o}}$. Fit γ_{o}, β_{o} using linear regression with $S(\phi)$
    $R^2[\tilde{\phi}(x)] \leftarrow$ (12)
    if $R^2[\tilde{\phi}(x)] > R^2_k$ then
      $\hat{\phi}_k(x) = \tilde{\phi}(x)$; $R^2_k = R^2[\tilde{\phi}(x)]$
    end
  end
  $Z[\hat{\phi}_k(x)] \leftarrow$ (14)
  if $Z[\hat{\phi}_k(x)] > Z_{\text{best}}$ then
    $\phi_{\text{sym}}(x) = \hat{\phi}_k(x)$; $Z_{\text{best}} = Z[\hat{\phi}_k(x)]$
  end
end
return $\phi_{\text{sym}}(x)$

˜给定 γ_{i}、β_{i}、γ_{o}、β_{o} 和 $f_k(x)$。对于每个 $\phi(x)$，我们计算 R^2 分数

$$R^2[\tilde{\phi}(x)] = 1 - \frac{\sum_{i=1}^N [\phi(x_i) - \tilde{\phi}(x_i)]^2}{\sum_{i=1}^N [\phi(x_i) - \bar{\phi}(x_i)]^2} \ , \tag{12}$$

¯其中 $\bar{\phi}(x_i) = \mathbb{E}[\phi(x_i)]$。接下来，我们设定

$$\hat{\phi}_k(x) = \arg\max_{\tilde{\phi}(x)} \ R^2[\tilde{\phi}(x)] \ , \tag{13}$$

ˆ其中，$\phi_k(x)$ 是给定 $f_k(x)$ 时对 $\phi(x)$ 的最佳近似。在基于 $\phi_k(x)$ 确定符号表达式 $\phi_{\text{sym}}(x)$ ˆ时，我们利用式 (11) 和式 (12) 构造ˆ组合评分项 $Z[\phi_k(x)]$„，其定义为

$$Z[\hat{\phi}_k(x)] = R^2[\hat{\phi}_k(x)] + \frac{\lambda}{Q[\hat{\phi}_k(x)]} \ . \tag{14}$$

此处，λ 为非线性评分项赋权。使用式 (14) 中的组合评分，我们计算

$$\phi_{\text{sym}}(x) = \arg\max_{\hat{\phi}_k(x)} \ Z[\hat{\phi}_k(x)] \ . \tag{15}$$

在本研究中，对于给定的 $\tilde{\phi}$（x），通过网格搜索确定使R^2 最大化的参数 γ_{i} 和 β_{i} 。此外，对于每个（γ_{i}，β_{i}）对，使用最小二乘回归确定 γ_{o} 和 β_{o}，其中 γ_{o} 和 β_{o} 分别是 S（ϕ）的最佳拟合斜率和截距。所述方法在 [5], 的基础上，加入了我们提出的非线性评分项，以便在可能的情况下促进节能方程的生成。算法1对SR过程进行了详尽概述。

C. 基于非线性度量对多层感知机和Kolmogorov-Arnold网络进行评分

对于给定的多层感知机网络，总非线性评分综合了线性激活函数和非线性激活函数各自的评分。因此，总评分 Q[MLP(x)] 表示为

$$Q[\text{MLP}(\mathbf{x})] = \sum_{l=1}^{L} \left[d_l \times \left(d_{l-1} + Q[\sigma^{(l)}(x)] \right) \right] \ , \tag{16}$$

其中，d_0 是第一层的输入大小。显然，每一层中 $\sigma^{(l)}$ 的选择都会影响总评分。

现在考虑一个 Kolmogorov-Arnold网络，其中 $\phi_i^{(l)}$ ，$_j$ 是连接第 i 个输入与第 j 个输出的第 l 层中的激活函数。总评分 Q[KAN(x)] 可通过分别处理每个学习得到的激活函数，并使用第三节A中描述的方法计算 $Q[\phi_{i,j}^{(l)}$ (x)] 来确定。在计算每个激活函数的评分时，我们考虑为 $\phi_i^{(l)}$ ，$_j$ 推导出的符号表达式，而非原始的 B样条实现。将 Kolmogorov-Arnold网络中所有激活函数的各项评分相加，可得

$$Q[\text{KAN}(\mathbf{x})] = \sum_{l=1}^{L} \sum_{j=1}^{d_l} \sum_{i=1}^{d_{l-1}} \left(a_{i,j}^{(l)} \times Q[\phi_{i,j}^{(l)}(x)] \right) \ . \tag{17}$$

这里，$a_i^{(l)}$,$_j$ 若 $\phi_i^{(l)}$,$_j$ 被剪枝，则取值为 0，否则为 1。剪枝过程详见第 III-D1 节。需要指出的是，对于 Kolmogorov-Arnold 网络，每个 $\phi_i^{(l)}$,$_j$ 的分数是在激活函数的网格区间上确定的。对于多层感知机，该区间可以根据每个隐藏层中激活函数的定义域、值域和有界性来选择。

D. 进一步改进的细节

1) 剪枝：为进一步提升 Kolmogorov-Arnold 网络的能效，我们采用了 [5]中的剪枝方法。对于具有多个层的 KAN，每个神经元的重要性由输入和输出分数决定

$$I_i^{(l)} = \max_k \left(||\phi_{i,k}^{(l-1)}(x)||_1 \right), \ O_i^{(l)} = \max_j \left(||\phi_{j,i}^{(l+1)}(x)||_1 \right) \ , \tag{18}$$

其中，$\phi_{i\,k}^{(l-1)}$,(x) 和 $\phi_{j\,i}^{(l+1)}$,(x) 分别表示连接到第 i 个神经元以及从该神经元连接出去的边上的激活函数，这些边位于第 l 层。输入和输出分数均高于阈值 η 的神经元会被保留，其余神经元则会被剪枝。对于 KAN层，我们还可以考虑剪除单个激活函数，而不是神经元。在这种情况下，对所有激活函数均考虑 $||\phi_{i\,k}^{(l-1)}$,(x)$||_1$ ，若该值低于 η, 则剪除对应的边。剪枝有助于通过移除冗余激活函数获得紧凑的闭式表达式，从而提升能效。

2）训练：为优化块错误率性能，我们采用噪声调度，并使用改进的交叉-

算法 2 带噪声调度的自编码器训练

输入： m, B, α, $\sigma^2_{\max}$, $\sigma^2_{\min}$, n_{epochs}
输出：优化后的参数 $\{\theta_{\text{e}}, \theta_{\text{d}}\}$
$\sigma_n^2 = \sigma^2_{\max}$
对于 $n = 1$ 到 n_{epochs}，执行
  采样 $\{m^{(i)}\}_{i=1}^{B}$ 来自 m
  $s_{\text{tx}} =$ 编码器$(m; \theta_{\text{e}})$
  $s_{\text{rx}} = s_{\text{tx}} + w$， $w \sim \mathcal{CN}(0,\ \sigma^2)$
  $\hat{y} =$ 解码器$(s_{\text{rx}}; \theta_{\text{d}})$
  $\mathcal{L} \leftarrow$ (19)
  使用 $\mathcal{L}$ 和 α更新 θ_{e} 和 θ_{d}
  $\sigma_n^2 = \sigma^2_{\max} - (n/n_{\text{epochs}}) \times (\sigma^2_{\max} - \sigma^2_{\min})$
end

熵损失函数

$$\mathcal{L} = -\sum_{i=1}^{2^k} T_i \log\left(\frac{\exp(l_i)}{\sum_{j=1}^{2^k}\exp(l_j)}\right), \tag{19}$$

其中， T_i 是第 i类的真实标签， l_i 和 l_j 分别是第 i个和第 j个解码器输出的对数优势。由于解码器输出端没有softmax层，因此模型直接输出对数优势。在本研究中，我们使用Adam优化器调整编码器参数θ_{e} 和解码器参数 θ_{d} ，以对编码器和解码器进行联合训练。算法 2概述了这一过程，其中 B 是批量大小， $\sigma^2_{\min}$ 和 $\sigma^2_{\max}$ 给出了噪声调度范围， α 是学习率。

IV. 数值结果

在数值实验中，我们将端到端OFDM系统中的(24,12)自编码器作为概念验证进行分析，并计划在未来工作中采用更大的分组大小。作为比较，我们考虑输入层、隐藏层和输出层各一个的多层感知机。隐藏层使用修正线性单元（ReLU）激活函数，而输出层不使用激活函数。在本研究中，我们在编码器和解码器中均采用具有150个隐藏层神经元的多层感知机。对于KAN自编码器，我们在编码器和解码器中均以单个KAN层替代多层感知机，其中每个可学习激活函数具有5个可学习控制点c ，并使用三次多项式基函数。为避免移除KAN自编码器的关键组件，我们在编码器处采用适度的剪枝阈值 $\eta = 10^{-4}$ ，在解码器处采用$\eta = 3 \times 10^{-5}$ 。在SR过程中，对于第三节A中所述的非线性评分计算，我们采用误差容限 $\epsilon = 10^{-2}$ ；对于第三节B中给出的非线性评分权重，则采用$\lambda = 3 \times 10^{-2}$ 。

我们使用 PyTorch 机器学习库，在 Python 中创建、训练并测试多层感知机自编码器和 KAN自编码器。使用 Adam 在加性高斯白噪声环境下训练每个模型 3×10^4 个训练轮次，每个批次包含 2^{11} 个随机选取的 m，学习率设为 $\alpha = 10^{-3}$。我们使用 $E_{\text{b}}/N_0 = 0$ dB 来计算 $\sigma^2_{\min}$ ，并使用 $E_{\text{b}}/N_0 = 6$ dB 来计算$\sigma^2_{\max}$。每个 KAN 激活函数的网格区间会定期更新，以适配训练样例。所有模型均使用 NVIDIA RTX 3070 GPU 进行训练。我们将多层感知机自编码器和 KAN自编码器与采用最大似然译码（MLD）的 (24,12) Golay码进行比较。我们对Golay码的 MLD 实现采用正交相移键控（QPSK）作为调制方案，其中 $\mathbb{E}[|s_{\text{tx}}|^2] = 1$。设 $\mathbf{s}_{\text{g}} \in \mathbb{C}^{n/2}$ 为构成调制码字的符号向量。我们将 MLD 实现为

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} \ \text{Re}\{\mathbf{s}_{\text{g}}^{\text{H}}\mathbf{c}\}\ , \tag{20}$$

其中，$\hat{\mathbf{c}}$ 是检测到的调制码字，而$\mathbf{c} \in \mathbb{C}^{n/2}$是一个包含Golay码QPSK调制码字的向量。对于该MLD实现，线性运算的数量为 $n^2 \times 2^k$，我们据此计算(24,12) Golay最大似然译码的非线性评分 2.359296×10^6 。

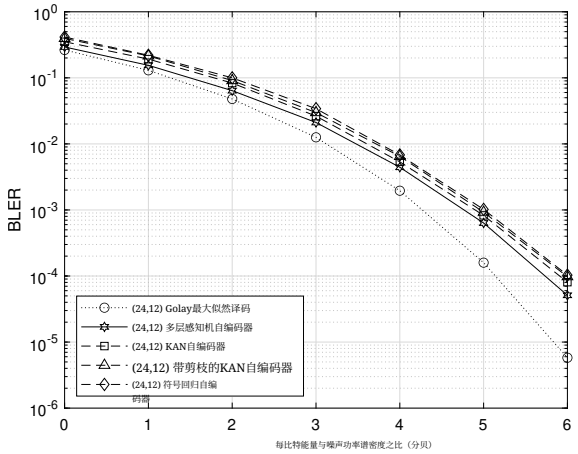
A. 误块率性能

为表征块错误率性能，我们进行了蒙特卡洛实验。首先，我们在加性高斯白噪声中仿真多层感知机自编码器和KAN自编码器的块错误率，并将其与(24,12) Golay最大似然译码进行比较。图2a中的误块率曲线显示，KAN自编码器的性能与多层感知机自编码器和(24,12) Golay最大似然译码相近，其中(24,12) Golay最大似然译码略优于前两者。值得注意的是，KAN自编码器和多层感知机自编码器的性能几乎完全相同。此外，在图2a中可以看到，剪枝对KAN自编码器整体块错误率性能的影响相对较小。我们还注意到，基于KAN的符号回归自编码器（符号回归自编码器）与剪枝模型相比，性能没有损失。

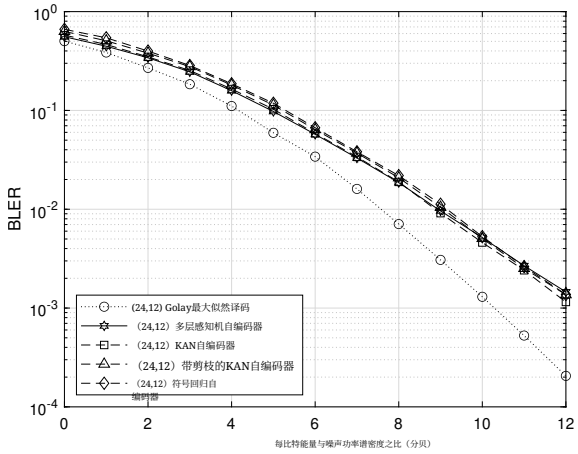
我们在瑞利衰落信道下进行了另一项蒙特卡洛实验，再次将 MLP-AE 和 KAN-AE 的块错误率与 (24,12) Golay 最大似然译码进行比较。在这项实验中，由于采用了正交频分复用，每个子载波上的符号均经历平坦衰落，并且我们假设接收机能够进行完美的信道估计。从图 2b 可以看出，所有模型的性能都相当。此外，与 AWGN 信道类似，剪枝会对块错误率性能产生轻微的负面影响，而 SR-AE 的性能与剪枝后的模型几乎完全相同。仿真结果表明，SR-AE 的性能与原始 KAN-AE 非常接近，这说明模型的准确率得以保持。

B. 功率与能耗

我们开展了另一项试验，其中使用 MLP-AE、SR-AE 和 Golay码处理 $5{\times}10^3$ 消息m ，试验次数固定为 2.5×10^4 。需要注意的是，该仿真包括编码器、信道和解码器。我们监测推理过程中的图形处理器功耗



（a）加性高斯白噪声中的误块率。



（b）瑞利衰落信道中的误块率。

图2. 不同（24,12）编码方案的性能。

并将每个模型的能耗计算为功耗曲线下方的面积。图 3 展示了各方案随时间变化的图形处理器功耗对比。由于本试验使用的是 NVIDIA RTX 3070 GPU，因此在所有情况下的平均功耗都非常高；然而，在无线电设备或移动设备等实际系统中，可以牺牲评估速度来显著降低功耗。此外，我们强调，图 3 中的曲线取决于具体实现和硬件。因此，使用非线性评分来评估各方案的相对能效更为恰当，因为该指标剔除了实现和硬件细节的影响。在图 3 中，我们观察到，与基于KAN的SR-AE相比，MLP-AE的能耗约高出 1.38 倍。这里需要指出的是，就能耗而言，Golay码的最大似然解码表现最佳，这可以归因于其实现所依赖的 PyTorch 库在硬件层面进行的优化。

MLP-AE 的峰值功耗和总能耗，以及非线性评分

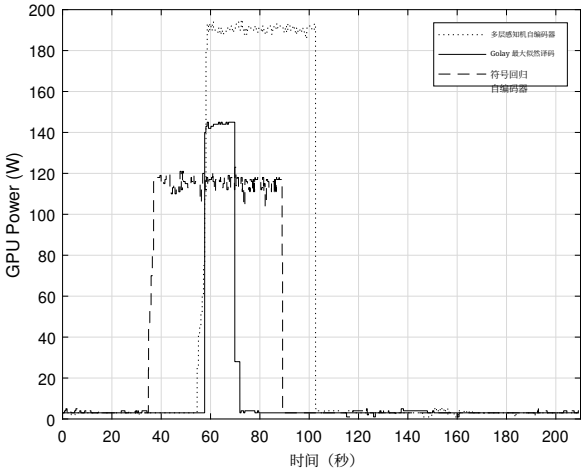


图3.（24,12）编码方案运行期间的图形处理器功耗。由于这是一种启发式测量，非线性评分能更好地衡量能效，且不受硬件和实现方式的影响。

表I 多层感知机自编码器 vs. 符号回归自编码器 vs. Golay 最大似然译码

	多层感知机自编码器	符号回归自编码器	Golay 最大似然解码
峰值功率	195 W	123 W	145 W
能耗	9166.9 J	6644.4 焦耳	2420.0 焦耳
非线性评分	1。2366e+6	6。8464e+5	2.3592e+6

和 SR-AE 的数据见表I。为计算 MLP-AE 的非线性评分，我们考察每个 ReLU 激活函数的单独评分。由于 ReLU 是具有 $N = 2$ 的分段线性函数，因此为每个隐藏层激活赋予 2 分。AE 中两个多层感知机的输出层均不包含激活函数，因此每个输出层激活函数赋予 0 分。然后，利用 (16) 计算出表I所示的评分。接下来，考虑经过剪枝并转换为符号表达式的 KAN-AE。每个激活函数都在其网格区间上进行考察；编码器中的激活函数对应的网格区间为 $[0, 1]$ ，解码器中的为 $[-2.2, 2.2]$ 。利用 (17) 计算出表I所示的 SR-AE 评分。

V. 总结性评论

本研究表明，在信道编码任务中，KAN相比多层感知机具有能效和模型大小方面的优势。KAN能够推导出符号表达式，因此可以得到简洁、低复杂度的方程，从而降低推理过程中的计算负担。为获得更简单的符号表达式，我们提出了一种非线性度量，并用其为不同的符号表达式评分，在符号回归过程中消除不必要的高非线性激活函数。结果表明，KAN自编码器在加性高斯白噪声信道和平坦衰落瑞利信道中的性能与多层感知机自编码器相近，同时结合所提出的符号回归方法还可降低能耗，

这使得Kolmogorov-Arnold 网络成为在实际通信系统中将深度学习模型集成到能量受限设备中的一种有前景的选择。未来的工作将着重于改进非线性度量、考察更大型的编码，并开展敏感性分析，以研究评分项对符号回归过程的影响。

参考文献

[1] M. Soltani、V. Pourahmadi、A. Mirzaei 和 H. Sheikhzadeh，“基于深度学习的信道估计”，IEEE Communications Letters，第 23 卷，第 4 期，第 652–655 页，2019 年。[2] M. Zhang、Y. Zeng、Z. Han 和 Y. Gong，“使用深度学习架构的自动调制识别”，载于 IEEE无线通信信号处理进展国际研讨会（SPAWC）会议论文集，2018 年，第 1–5 页。[3] Q. Mao、F. Hu 和 Q. Hao，“面向智能无线网络的深度学习：综合综述”，IEEE Communications Surveys & Tutorials，第 20 卷，第 4 期，第 2595–2621 页，2018 年。[4] H. Huang、S. Guo、G. Gui、Z. Yang、J. Zhang、H. Sari 和 F. Adachi，“面向物理层 5G 无线技术的深度学习：机遇、挑战与解决方案”，IEEE Wireless Communications，第 27 卷，第 1 期，第 214–222 页，2020 年。

[5] Z. Liu、Y. Wang、S. Vaidya、F. Ruehle、J. Halverson、M. Soljačić、T. Y. Hou 和 M. Tegmark，“KAN：Kolmogorov-Arnold网络”，arXiv 预印本 arXiv:2404.19756，2024 年。[6] R. Yu、W. Yu 和 X. Wang，“KAN 还是多层感知机：更公平的比较”，arXiv 预印本 arXiv:2407.16674，2024 年。[7] Z. Liu、P. Ma、Y. Wang、W. Matusik 和 M. Tegmark，“KAN 2.0：Kolmogorov-Arnold网络与科学相遇”，2024 年。

[ ] 2408 10205 在线。可用地址：[8] https://arxiv.org/abs/ 。C. J. Vaca-Rubio、L. Blanco、R. Pereira 和 M. Caus，“用于时间序列分析的 Kolmogorov-Arnold 网络（KANs）”，arXiv 预印本 arXiv:2405.08790，2024 年。[9] T. O’Shea 和 J. Hoydis，“物理层深度学习简介”，IEEE Transactions on Cognitive Communications and Networking，第 3 卷，第 4 期，第 563–575 页，2017 年。[10] A. Felix、S. Cammerer、S. Dörner、J. Hoydis 和 S. Ten Brink，“用于通信系统端到端学习的 OFDM自编码器”，载于 IEEE无线通信信号处理进展国际研讨会（SPAWC）会议论文集，2018 年，第 1–5 页。[11] D. Wu、M. Nekovee 和 Y. Wang，“基于深度学习的自编码器用于多用户无线干扰信道物理层设计”，IEEE Access，第 8 卷，第 174 679–174 691 页，2020 年。[12] A. K. Kolmogorov，“通过连续单变量函数的叠加与相加表示多个变量连续函数”，苏联科学院院报，第 114 卷，第 369–373 页，1957 年。