将形式逻辑蒸馏到神经空间：用于信号时序逻辑的核对齐方法

Sara Candussio SARA.CANDUSSIO@PHD.UNITS.IT
AILab，MIGe，的里雅斯特大学，意大利
GabrieleSarti G.SARTI@NORTHEASTERN.EDU库里
计算科学学院，东北大学，美国马萨诸塞州波士顿
Gaia Saveri GAIA.SAVERI@PHD.UNITS.IT
AILab，MIGe，的里雅斯特大学，意大利
Luca Bortolussi LBORTOLUSSI@UNITS.IT
AILab，MIGe，的里雅斯特大学，意大利

编辑： NeSy 2026审稿中

arXiv:2603.05198v1 [cs.CL] 5 Mar 2026

摘要

我们提出了一种学习形式规范连续神经表征的框架，通过将其语义的几何结构提炼到潜在空间中来实现这一目标。现有方法要么依赖符号核——这种方法能够保留行为语义，但计算代价高昂、依赖锚点且不可逆——要么依赖基于语法的神经嵌入，而后者无法捕捉底层结构。我们的方法弥合了这一差距：通过采用教师-学生架构，我们将符号鲁棒性核提炼到 Transformer 编码器中。不同于标准的对比方法，我们使用连续的核加权几何对齐目标来监督模型，使其按照语义差异的大小对错误进行惩罚。训练完成后，编码器能够通过单次前向传播生成嵌入，以远低于核计算的成本有效模拟其逻辑。我们将该框架应用于信号时序逻辑（STL），结果表明，所得神经表示能够忠实保留 STL 公式的语义相似性，准确预测鲁棒性和约束满足情况，并且本质上保持可逆性。我们提出的方法无需在运行时间反复计算核，即可实现高效、可扩展的神经符号推理和公式重构。

1. 引言

诸如信号时序逻辑（STL，Maler与Ni˘ckovi´c，2004年）等形式规约语言，是信息物理系统分析与控制的核心，能够将针对时变信号的安全性、活性与性能要求加以形式化。近期研究将STL融入机器学习流程（Raman等人，2015年；Bombara与Belta，2021；Akazaki等人，2018年；Li等人，2017年；Hasanbeig等人，2020年），从而支持需求挖掘、规范引导学习与神经验证等任务。STL借助鲁棒性给出定量语义（Fainekos与Pappas，2006年），用以测度公式被满足或违反的程度。基于鲁棒性的核方法提供了一种语义层面的替代方案：通过在信号分布上计算鲁棒性来比较公式（Bortolussi等人，2022年），并将规约嵌入希尔伯特空间，其中距离反映的是行为上的一致而非语法差异（Saveri等人，2024年）。然而，估计成对相似度需要反复进行鲁棒性评估，并且其规模随

© 2026年 S. Candussio、G. Sarti、G. Saveri 和 L. Bortolussi。

公式数量呈二次增长。此外，行为等价但词法不同的规约会坍缩到同一点，使该映射不可逆。这对相似度估计尚可接受，但在生成或操作大量公式时则存在问题。尽管已有研究探索了从核嵌入进行神经重构（Candussio等人，2025），但推广这些结果关键取决于为模型训练所选公式参考集的覆盖范围。

近期研究探索了使用基于 Transformer 的神经编码器将公式嵌入共享向量空间，从而生成可通过一次前向传播高效计算的表示（Allamanis 等人，2018年；Feng 等人，2020年；Dong 等人，2021；Reimers 和 Gurevych，2019年）。然而，现有方法大多主要依赖词汇相似性（Gao 等人，2022年），导致嵌入无法充分捕捉形式语言规范所蕴含的复杂语义关系。

我们通过训练神经编码器来近似 STL 语义核所诱导的几何结构，从而弥合这两种范式。我们将该核视为教师度量，并将表示学习构建为关系蒸馏过程（Hinton 等人，2015年；Park 等人，2019年），使嵌入内积与基于核鲁棒性的相似度对齐。Transformer 编码器使用成对核相似度进行训练，采用加权几何对齐目标，重点校正较大的语义差异。由此，鲁棒性计算的成本被前置到训练阶段，使用蒸馏模型进行推理时仅需一次前向传播，从而摊销评估计算的成本。

贡献

1. 我们提出一种将 STL 定量语义提炼为神经嵌入的方法，并采用基于鲁棒性的语义核来控制学习所得表征的几何结构。由此得到的潜在空间近似于相应的再生核希尔伯特空间（再生核希尔伯特空间（Berlinet 和 Thomas-Agnan，2004年），并依据行为语义组织公式。

2. 我们提出一种加权成对目标，促使神经编码器模仿核相似度。该加权方式优先处理模型与核信号偏差最大的样本，确保编码器聚焦于其最大误差。

2. 背景

2.1. 信号时序逻辑

信号时序逻辑（STL）（Maler 和 Ničkovi´c，2004年）是一种线性时序逻辑，用于描述稠密时间域上实值轨迹的性质。轨迹（或信号）是一个函数 $\xi : I \to D$，其中 $I \subseteq \mathbb{R}_{\geq 0}$ 是一个时间区间，$D \subseteq \mathbb{R}^k$ 是状态空间。我们用 $\mathcal{T}$ 表示容许轨迹的空间，用 $\mathcal{F}$ 表示良构 STL 公式的集合。STL 的语法定义为 $\varphi ::= \top \mid \pi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathrm{U}_{[a,b]} \varphi_2$，其中 π 是形如 $f_\pi(x) \geq 0$ 的原子谓词，$\neg$ 和 $\wedge$ 是布尔联结词，而 $\mathrm{U}_{[a,b]}$ 是有界时域直到算子；由此可导出最终 $\mathrm{F}_{[a,b]}\varphi \equiv \top \mathrm{U}_{[a,b]} \varphi$ 和始终 $\mathrm{G}_{[a,b]} \equiv \neg \mathrm{F}_{[a,b]} \neg\varphi$ 算子。

STL 同时支持布尔与定量语义（附录A）；布尔语义决定轨迹 ξ 是否满足公式 φ 在时刻 t；定量

语义称为鲁棒性，记作 $\rho(\varphi, \xi, t) \in \mathbb{R}$，用于衡量时刻 t 满足或违反的程度。下文中，当 $t = 0$ 我们使用简写表示法 $\rho(\varphi, \xi)$ 时，表示 $\rho(\varphi, \xi, 0)$。正鲁棒性意味着满足，负鲁棒性意味着违反，而其幅值量化了与满足边界的距离。

2.2. 信号时序逻辑的语义保持嵌入

定量语义为规约提供了一种函数视角：每个公式 $\varphi \in \mathcal{F}$都可以解释为定义在轨迹上的实值函数，$\Phi(\varphi) := \rho(\varphi, \cdot)$。固定关于 $\mathcal{T}$ 的概率测度 μ_0 后，可由此得到表示 $\Phi(\varphi) \in L^2(\mathcal{T}, \mu_0)$，并通过内积定义语义相似度量：

$$k(\varphi_i, \varphi_j) = \langle \Phi(\varphi_i), \Phi(\varphi_j) \rangle_{\mu_0} = \int_{\mathcal{T}} \rho(\varphi_i, \xi)\rho(\varphi_j, \xi)\, d\mu_0(\xi). \tag{1}$$

这一构造在 $\mathcal{F}$ 上定义了一个正定核，并将 STL 规范嵌入再生核希尔伯特空间（RKHS）中。在该空间里，接近程度反映的是轨迹分布 μ_0 上的行为相似性，而非词汇匹配。

实际应用中，积分通过对轨迹进行蒙特卡洛采样来近似。通常还会应用余弦归一化和指数映射，以获得尺度不变的 RBF 核（附录 A）。由此得到的几何结构具有行为层面的解释：两个公式嵌入之间的距离衡量了期望鲁棒性上的差异，从而提供了一种植根于系统行为的连续语义相似性概念。

2.3. 核嵌入反演与覆盖限制

根据 STL 核的定义（公式 1），固定一组（可能是随机选取的）D 锚点公式$\{\varphi_j\}_{j=1}^{D}$，并构造向量 $k(\varphi) = [k(\varphi,\varphi_j)]_{j=0}^{D-1}$，即可得到公式 $\varphi \in \mathcal{F}$ 的连续表示 $k(\varphi) \in \mathbb{R}^D$。因此，如果我们希望将另一个规范 $\varphi' \in \mathcal{F}$ 映射到同一潜在空间，以便对所得嵌入进行有意义的比较，就需要依靠同一锚集来获得 $k(\varphi') = [k(\varphi',\varphi_j)]_{j=0}^{D-1}$。因此，所得表示并非公式本身固有的，而是相对于由锚集诱导的坐标系定义的。基于鲁棒性的此类嵌入的反演已通过对中等复杂度公式进行大规模最近邻检索（Saveri 和 Bortolussi，2024）以及在 $\sim 10^5$ 示例上训练 Transformer 解码器（Candussio 等，2025）进行了探索。这两种方法都依赖有限的锚集，该锚集界定了它们所能表示的语义空间范围。

2.4. 神经编码器与池化策略

原则上，可以用神经网络学习符号表达式的嵌入（Al- lamanis 等人，2018年；Feng 等人，2020年）。参数化编码器 f_θ： $\mathcal{S} \to \mathbb{R}^d$ 可以训练为将每个符号对象 $s \in \mathcal{S}$ 映射到一个固定维度的向量。可以使用 Transformer 处理符号表达式（Vaswani 等人，2017年），生成能够捕捉局部结构和长程依赖的词元嵌入。为了获得表示整个表达式的单个嵌入，可以通过池化聚合词元级表示。典型的

策略包括对词元表示求平均（均值池化），或使用指定词元的表示（例如，特殊序列起始词元 [CLS]和 [BOS]），该词元经过专门训练，用于概括序列并捕捉其全局上下文信息（Devlin 等人，2019年；Radford 和 Narasimhan，2018年；Reimers 和 Gurevych，2019年）。

3. 相关工作

3.1. 核引导的神经表示学习

在核蒸馏和度量监督的框架下，学习能够保留目标核几何结构的神经嵌入已得到研究。核蒸馏和度量监督在这一框架中，核定义输入之间的成对相似度，神经编码器则经过训练，在其潜在空间中复现这些相似度（Park 等人，2019年）。表示蒸馏方法已在多个领域得到提出，例如卷积网络（Jones 等人，2019）和神经切线核（Mahowald 等人，2026年）。理论分析表明，在超球面上进行保持相似度的学习能够近似核算子的主特征子空间（Lee 等人，2025）。因此，连续的核相似度可充当软监督，使神经编码器能够内化核所诱导的语义几何，并实现对底层再生核希尔伯特空间的有限维近似。与对比目标不同，这种方法强制要求与语义结构保持全局一致，而非仅进行相对判别（Zbontar 等人，2021）。

3.2. 用于符号表示的神经架构和投影头

在现代自监督对比学习框架中，例如 SimCLR（Chen 等人，2020年）、BYOL（Grill 等人，2020年）、MoCo v2（Liu 等人，2024年），将结构编码与度量对齐解耦是一种成熟的策略。这些框架通常采用深层骨干网络进行语法感知特征提取，并通过可学习投影头将这些特征映射到目标潜在空间，以进行度量比较。受这些技术启发，我们采用类似的分离方式，使神经结构表示与鲁棒性核所诱导的连续语义几何对齐，同时确保骨干网络保留逻辑组合性，并由投影头确保语义一致性。

4. 方法

我们的目标是学习一个神经编码器 $f_\theta : \mathcal{F} \to \mathbb{S}^{D-1}$，其中 D 是嵌入空间的维数，该编码器将 STL 规范 $\varphi_i,\ \varphi_j \in \mathcal{F}$ 映射为潜在表示 e_i, e_j，使其点积近似于公式 1中的 STL 核相似度：

$$\langle e_i, e_j \rangle \approxeq k(\varphi_i, \varphi_j)$$

尽管核 $k(\cdot,\ \cdot)$ 隐式定义了一个映射，将其映入（理论上）无限维的再生核希尔伯特空间，但编码器 $f_\theta(\cdot)$ 学习到一个有限的 D维表示，用以近似该空间中的成对语义关系。这通过加权几何对齐损失 实现（见第 4.1 节），该损失在 STL 核下促进语义一致性。因此，语义相似性高的公式会被引导至

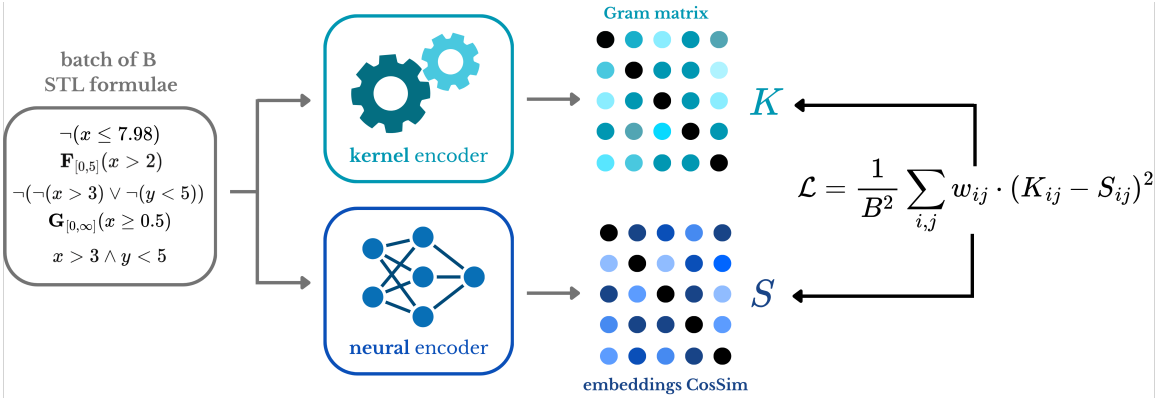


图 1：我们提出的方法概览：将鲁棒性语义从符号 STL 核蒸馏到基于 Transformer 的编码器中，以实现高效推理。

在超球面上彼此靠近，而语义上不相似的公式则会被推向近乎正交的位置。

4.1. 加权几何对齐损失

训练过程将对潜在表示 z（公式 φ 由基于 Transformer 的编码器 $f_\theta(\cdot)$ 生成）与语义核 $k(\varphi, \varphi_{ref})$ 所诱导的几何结构对齐。与使用二元标签的标准对比方法不同[1]，，我们的框架将核值 $K_{ij} = k(\varphi_i, \varphi_j)$ 视为回归任务中的连续目标。

给定一个公式批次 B，我们将核加权对比损失定义为：

$$\mathcal{L} = \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} w_{ij} \cdot (K_{ij} - S_{ij})^2$$

其中 $K_{ij} = k(\varphi_i, \varphi_j) \in [0, 1]$ 为 STL 核相似度，而 $S_{ij} = \langle e_i, e_j \rangle$ 为 L_2 归一化嵌入 $e = f_\theta(\varphi) / \|f_\theta(\varphi)\|$ 之间的点积。

虽然余弦相似度 S_{ij} 的取值范围是 $[-1, 1]$，但核 K_{ij} 是非负的（2.2，附录A）。这种表面上的差异可以从几何角度解释：语义无关的公式对（$K_{ij} \approx 0$）会被推向超球面上的正交状态（$S_{ij} \approx 0$），从而产生最大的逻辑分歧，对应于零相似度。这确保潜在几何与 STL 核的概率一致性表述保持一致。

权重w_{ij} 充当动态焦点机制，优先关注 Transformer 的结构理解与核语义偏差最大的公式对：

$$w_{ij} = \min\left(\frac{|K_{ij} - S_{ij}|^\gamma}{\mathbb{E}_{i,j \in B}[|K_{ij} - S_{ij}|^\gamma]}, C\right)$$

$|K_{ij} - S_{ij}|^\gamma$ 会放大表现出高度语义错位的配对的梯度，而$\mathbb{E}_{i,j \in B}[|K_{ij} - S_{ij}|^\gamma]$ 则将该量相对于当前批次平均值进行归一化

1. 公式对分为正例（匹配）和负例（不匹配）。

误差，从而稳定学习信号。钳位操作将任意配对的最大允许影响限制为常数 C，从而防止异常值引起梯度不稳定性。参数 γ 控制聚焦机制，较高的取值会迫使编码器更快地在困难配对上收敛。该加权不影响目标几何结构，但会在训练期间调节梯度分配。

4.2. 编码器模块

编码器 $f_\theta(\varphi)$ 使用基于深度 Transformer 的特征提取器和非线性 MLP 投影头，将离散的 STL 语法映射到连续的潜在空间中。

Transformer 编码器 为捕捉 STL 规范中固有的层级依赖和长程依赖，我们采用了一个 12 层 Transformer，每层包含 16 个注意力头。由于 STL 公式的语义取决于层级结构，我们采用可学习的位置嵌入来编码运算符的相对优先级。这使模型能够有效区分符号相同但语义不同的排列。[2] 相反，对于保持语义不变的排列，例如逻辑运算符的交换律，模型应保持不变。

池化 基于 Transformer 的分类器通常使用专门指定的 token（例如 `[CLS]`或 `[BOS]`），通过端到端训练将序列信息聚合为瓶颈表示（Devlin 等人，2019年；Radford 和 Narasimhan，2018年；Reimers 和 Gurevych，2019年）。我们评估了三种池化策略，以便在将 Transformer 的输出传递到最终投影层之前对其进行汇总：均值、 `[CLS]` 和 `[BOS]`池化（2.4）。

具体而言，令 $H \in \mathbb{R}^{N\times D}$ 表示最终 Trans- former 层的隐藏状态序列，其中 N 为序列长度， D 为嵌入维度，$M \in \{0,1\}^N$ 为二进制注意力掩码。对于 [CLS] 和 [BOS] 池化，直接取第一个标记作为表示。对于均值池化，使用注意力掩码确保只纳入有效的标记位置，并忽略填充。

$$e_{\texttt{[CLS]}} = e_{\texttt{[BOS]}} = H_1 \in \mathbb{R}^D. \qquad e_{\text{mean}} = \frac{\sum_{i=1}^N H_i \cdot M_i}{\sum_{i=1}^N M_i} \in \mathbb{R}^D \tag{2}$$

MLP 投影器 通过上述某种策略池化得到的嵌入 e经由具有瓶颈架构的两层多层感知机映射到最终潜在流形 $\mathcal{Z}$：

$$z = \text{MLP}(e) = W_2 \cdot \text{LayerNorm}(\sigma(W_1 \cdot e + b_1)) + b_2,$$

其中， σ 是 GELU 激活函数（Hendrycks 和 Gimpel，2016年），它能够提供平滑的非线性，特别适用于 STL 鲁棒性的连续梯度。

第一个线性层 $W_1 \in \mathbb{R}^{D\times D/2}$ 将 $D = 1024$ 维输入投影到 $D/2 = 512$ 维瓶颈，迫使编码器压缩重现 STL 核相似度所需的信息，从而在保留语义结构的同时过滤无关细节。在最终投影前应用层归一化（LayerNorm），以稳定特征分布。

2. 例如，always eventually φ 表示 φ 会发生 *infinitely often*（反复发生），而 eventually always φ 表示 φ 将持续保持为 *always true from a certain point on*（持续性）。

第二层 $W_2 \in \mathbb{R}^{D/2\times D}$ 映射回目标嵌入维度，并将输出进行 L_2 归一化，使其位于单位超球面 $\mathbb{S}^{D-1}$ 上：

$$e = \frac{z}{\|z\|} = \frac{f_\theta(\varphi)}{\|f_\theta(\varphi)\|}.$$

将 e 限制在流形 $\mathcal{Z} \subset \mathbb{S}^{D-1}$ 上可确保点积 $\langle e_i, e_j\rangle$ 直接计算公式之间的余弦相似度，使潜在空间能够忠实地编码定量核语义。同时，超球面几何可防止嵌入空间发生坍缩，并促进所有维度得到均匀利用，从而在保留语义关系的同时维持表达能力。

5. 实验设置

数据集构成 训练集和测试集由以下种子生成：Candussio等人(2025)，并采用附录 B中详述的增强流水线进行扩充。对于每个种子公式，我们生成：(i) 语义等价但词汇结构复杂的公式（10.4%）；(ii) 改变阈值数值和时间边界、同时保持结构不变的参数扰动（43.4%）；(iii) 兼具结构和语义变化的混合条目（45.7%）。这种分层方式确保模型能够学习捕捉不同句法和定量修改下的语义相似性，最终共得到 330 万个公式。

训练设置 第 4 节所述的架构使用核对齐损失（第 4.1 节）进行训练。使用 AdamW（Loshchilov和Hutter，2017年）进行优化，学习率为 $1 \cdot 10^{-5}$，采用 `bf16` 混合精度，并通过 4 步梯度累积，使有效批大小达到 512。公式按照 Candussio等人(2025)[3], 中的方法进行分词，最大序列长度为 512 个 token。

STL 语义核在前向传播期间按小批量在线计算，避免了预先计算完整 Gram矩阵所带来的巨大内存开销，同时使损失函数能够作用于当前嵌入。训练共进行 5 个训练轮次，同时监控核对齐情况以及嵌入在超球面上的分布。

训练指标我们通过衡量学习到的表示对 STL 语义核几何结构的复现忠实度，来量化其质量。由于该核在希尔伯特空间中诱导出内积（第2.2节），正确的嵌入应同时保留成对相似度和表示的整体分布。我们报告三项互为补充的指标：核加权语义损失（第4.1节）、核对齐以及嵌入的均匀性。核对齐是向量化后的核矩阵与神经相似度矩阵之间的余弦相似度，取值范围为 0 到 1，数值越高表示语义结构保留得越好。均匀性（Wang and Isola，2022年）用于衡量嵌入在超球面上的分布均匀程度，其取值为 $[-4, 0]$；0 表示完全坍缩，而 -4 则对应最大程度的均匀分布和语义内容的最佳分离（定义见附录C ）。

3. 在 [CLS] 场景中，添加一个特殊控制标记作为表示瓶颈。

6. 结果

6.1. 核蒸馏遵循度

采用上述定义的训练指标，我们发现三种池化策略（mean、[CLS]，和 [BOS]）均收敛至较高的符合度，证明了所提出蒸馏框架的鲁棒性。在各种配置下，核对齐始终超过 0.9，表明学习到的表示能够准确反映核相似度。我们还发现，均匀性指标稳定在 -3.0 左右，这表明公式嵌入分布于超球面上，从而避免维度坍缩。

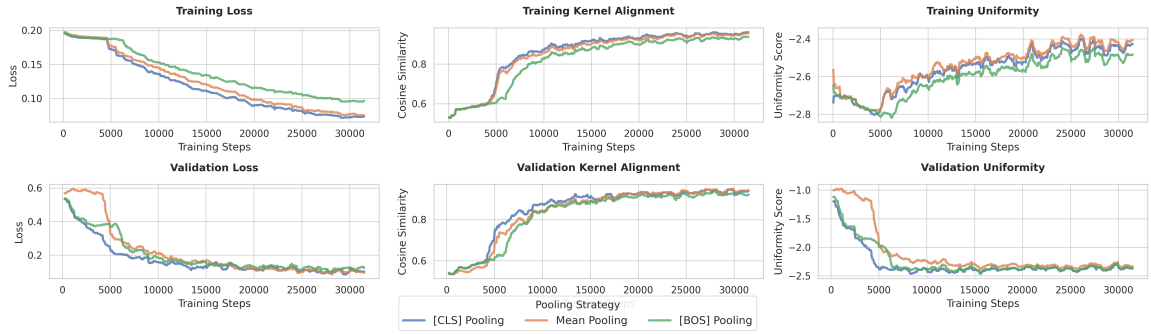


图2：三种池化策略（[CLS]，mean 和 [BOS]）在约 30,000 个训练步数中的训练（上）和验证（下）指标变化。

图2 的左侧和中间面板展示了几何对齐损失持续降低以及核对齐相应提升（> 0.9），这表明编码器有效地内化了目标语义几何结构。验证指标与训练表现高度一致，证实模型对未见公式具有很强的泛化能力，且没有发生句法过拟合。右侧面板显示，验证集上的均匀性得分稳定在-2.4 左右，表明这些表示维持了分布良好的超球面潜在空间，避免了维度坍缩。在所有配置中，[CLS]池化（蓝色）始终表现出最快且最稳定的收敛。

6.2. 语义一致性

为评估训练后的编码器能否恢复 STL 核语义，我们在逻辑等价公式对的测试集（≈ 3000 个示例）上，将神经余弦相似度与相应的 STL 核值进行比较。对于每一对（φ_1，φ_2），我们计算神经指标与基于核的指标之间的平均绝对误差（MAE）；数值越低，表示对齐效果越好。作为基线，我们采用测试集中的随机公式（不等价），以及词汇相似但语义不同的公式，即选取编辑距离和核相似度最小的公式 < 0.7 （词汇相似）。为便于直观比较不同公式类别，神经距离和核距离均按各自的最大值进行归一化。

如表 1所示，模型为等价对赋予了较高的相似度（0.966），为不等价对赋予了较低的相似度（0.182）。至关重要的是，它能够抵御词汇相似 困难负样本

表1： [CLS]池化的语义一致性评估。

度量	等价	不等价	词汇相似
神经相似度	0.966	0.182	0.308
核相似度	0.997	0.170	0.225
平均绝对误差（语义差距）	0.034	0.072	0.112
相对神经距离	0.137	0.833	0.770
相对核距离	0.105	0.830	0.775

（0.308），这证明编码器成功地将离散语法映射到连续空间，忠实反映了真实的逻辑语义。

6.3. 效率分析

我们测量为每个公式计算向量表示所需的嵌入时间，以及比较两个已嵌入公式所需的相似度计算时间，以此捕捉相似度测量的不同可扩展性特征。

嵌入时间 STL 核的复杂度为 $\mathcal{O}(BNP)$，其中 B 表示公式的数量，N 表示蒙特卡罗评估中的信号数量，而 $P = 1000$ 表示估计点的数量（附录 A）。神经编码器的复杂度为 $O(BL^2)$，其中 L表示固定序列长度。尽管这些趋势可直接由理论复杂度推得，表2 中的结果表明，核的内存占用随 NP线性增长，而 Transformer 嵌入则不受 NP 影响；在所有测试的信号数量下，其运行时间和内存效率均显著提升（见附录 D）。

表2：在 NVIDIA A100 80GB GPU 上的嵌入效率比较（$B = 2000$）。loaded：预加载模型的推理时间；full 还包括模型加载时间。

度量	核						Transformer	
	$N = 500$	1000	2000	4000	8000	16000	已加载	full
时间（秒）	2.18	3.03	4.77	8.36	17.08	48.86	2.17	3.56
随机存取存储器（GB）	4.91	8.74	16.46	31.70	62.38	123.38	1.54	1.71

成对相似度计算 计算一个批次中 B公式的成对相似度需要构造一个 $B \times B$ 矩阵。然而，每个矩阵元素的计算成本存在显著差异。STL 核需要计算 N个采样信号上的鲁棒性向量之间的内积，每个信号包含 P个点，因此复杂度为 $\mathcal{O}(B^2NP)$。相比之下，神经编码器会生成固定维度嵌入 $e \in \mathbb{S}^{D-1}$，成对相似度可通过一次矩阵乘法计算，复杂度为 $\mathcal{O}(B^2D)$，并且能够高效地进行 GPU实现。由于通常 $D \ll N \cdot P$，尽管在 B 上的缩放仍为二次方，神经网络方法的延迟仍显著更低（见附录 D中的图）。

6.4. 探究平均鲁棒性与满足

为验证学习得到的嵌入是否捕捉了超越结构线索的定量 STL 语义，我们冻结编码器，并在 L_2-

归一化表示之上训练一个轻量级回归器，以预测每个公式 φ 的两个标量：平均鲁棒性 $\bar{\rho}(\varphi)$ 和满足概率 $p_{\text{sat}}(\varphi)$。这两个量均基于从轨迹上的固定概率分布 μ_0 中采样的 1000 个信号计算。作为参照，我们使用 STL 核格拉姆矩阵的行作为输入特征，重复相同的实验。在这两种设置下，我们都使用均方误差损失训练相同的两层多层感知机，以确保神经表示与基于核的表示[4]在架构上对称。我们使用皮尔逊相关系数 r 和平均绝对误差 (MAE) 评估模型在未见过的公式上的表现。其中，预测值 $\hat{y}$ 是轻量级探针多层感知机作用于嵌入后得到的输出，真实目标值 y 则是每个公式基于 1000 个采样信号计算得到的量：平均鲁棒性 $\bar{\rho}(\varphi)$ 或满足概率$p_{\text{sat}}(\varphi)$。神经嵌入在鲁棒性预测上的相关系数为 $r = 0.910$，MAE 为 $= 0.399$；在满足概率预测上的相关系数为$r = 0.947$，MAE 为 $= 0.062$。相比之下，使用核特征时，这两项指标分别为 $r = 0.999$ 和 MAE $= 0.064$，以及$r = 0.994$ 和 MAE $= 0.023$。较高的相关系数和较低的 MAE 表明，学习得到的嵌入空间保留了 STL 定量语义的大部分信息，同时提供了紧凑得多且更高效的表示。

6.5. 解码神经嵌入

为解决嵌入逆转问题（第2.3节），我们研究了能否使用类似于 Candussio等人（2025）所用的仅解码器Transformer，直接从冻结的神经表示中重建原始符号公式。为明确检验所学习潜在空间的结构与语义丰富性，我们特意将解码器的训练限制为 5 个训练轮次。[5] 如果连续嵌入忠实地捕捉了公式语义，下游解码器进行逆转所需的训练预算应显著减少。与前文相同，我们通过原始公式与解码公式的鲁棒性向量 $\rho(\varphi) \in \mathbb{R}^M$来测量二者的语义差异；这些向量是通过取一组 M 信号 $\Xi = \{\xi_i\}_{i=1}^M$，从 μ_0 作为 $\rho(\varphi)_i = \rho(\varphi, \xi_i)$ 采样得到的。直观而言，如果两个公式的鲁棒性向量相似，就意味着这些规格在 Ξ中的轨迹上表现相似，从而能够快速近似公式之间的语义相似性。即使在严格的计算预算下，基于冻结表示训练的解码器仍达到 0.8688 的余弦相似度中位数和 2.1766 的 L_2距离。作为参考，Candussio等人（2025）的基线报告的余弦相似度中位数为 0.926–0.985，L_2距离中位数为 12–28。尽管由于缩放方式不同，距离较小，但较高的余弦相似度表明，我们的嵌入捕捉到了大部分语义信息，因此只需远少于基线的训练步数即可高效完成逆转。

7. 结论

本文提出一种框架，将基于鲁棒性的核的语义提炼到基于 Transformer 的模型中，并展示了该框架在信号时序逻辑（STL）公式上的应用。标准核虽然能够准确比较行为，但计算开销大、依赖锚集，且本质上不可逆。我们

4. 该基线与经典的核岭回归不同：我们将参数化回归器应用于核派生特征，而非求解再生核希尔伯特空间回归问题的闭式解。5. 这相当于 Candussio 等人 Candussio等人 (2025) 的参考基线的四分之一。

证明，我们的方法能够以极低的运行时间开销生成嵌入，同时忠实保留语义相似性。

我们还证明，所得嵌入不仅具有丰富的语义信息，而且是可逆的：用于逆转核嵌入的解码器能够快速学会从蒸馏潜在空间中重构原始公式，表明该嵌入捕获了有关底层逻辑的丰富结构与语义信息。总而言之，我们的框架为 STL 规范的推理提供了一种实用且可扩展的方法，连接了形式化符号表示与连续向量嵌入，同时保留了公式比较、检索和重构的能力。未来的工作将探索把这种蒸馏方法扩展到多种基于字符串的核，并将其范围从时序逻辑拓展出去，以检验我们方法的普适性。

参考文献

Takumi Akazaki、Shuang Liu、Yoriyuki Yamagata、Yihai Duan 和 Jianye Hao。使用深度强化学习对信息物理系统进行伪造，第 456–465 页。Springer International Publishing，2018年。ISBN 9783319955827。doi：10.1007/ 978-3-319-95582-7 27。URLhttp://dx.doi.org/10.1007/978-3-319-95582-7_27。

Miltiadis Allamanis、Marc Brockschmidt 和 Mahmoud Khademi。学习用图表示程序，2018年。网址https://arxiv.org/abs/1711.00740。

Alain Berlinet 和 Christine Thomas-Agnan。概率与统计中的再生核希尔伯特空间。2004年1月。ISBN 978-1-4613-4792-7。doi：10.1007/978-1-4419-9096-9。

Giuseppe Bombara 和 Calin Belta。使用决策树离线和在线学习信号时序逻辑公式。*ACM*《信息物理系统汇刊》，5(3)，2021年3月。ISSN 2378-962X。doi：10.1145/3433994。网址https://doi.org/10.1145/3433994。

Luca Bortolussi、Giuseppe Maria Gallo、Jan K˘ret´insk´y 和 Laura Nenzi。随机过程上的信号时序逻辑模型检测学习与核技巧，2022年。网址https://arxiv.org/abs/2201.09928。

Sara Candussio、Gaia Saveri、Gabriele Sarti 和 Luca Bortolussi。连接逻辑与学习：通过 *Transformer* 解码时序逻辑嵌入，第 3–18 页。Springer Nature Switzerland，2025年 9 月。ISBN 9783032060969。doi：10.1007/978-3-032-06096-9 1。URL http://dx.doi.org/10.1007/978-3-032-06096-9_1。

Ting Chen、Simon Kornblith、Mohammad Norouzi 和 Geoffrey Hinton。视觉表征对比学习的简单框架。载于 第 *37* 届国际机器学习会议论文集，ICML'20。JMLR.org，2020年。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：面向语言理解的深度双向 Transformer 预训练。载于 计算语言学协会北美分会，2019年。URLhttps://api.semanticscholar.org/CorpusID:52967399。

Lijun Dong、Hong Yao、Dan Li、Yi Wang、Shengwen Li 与 Qingzhong Liang。通过基于复杂网络的锚结构改进图神经网络。*Knowledge-Based Systems*，233:107528，2021。ISSN 0950-7051。doi: https://doi.org/10.1016/j. knosys.2021.107528。URLhttps://www.sciencedirect.com/science/article/pii/S0950705121007905。

Georgios E. Fainekos 与 George J. Pappas。时序逻辑规约的鲁棒性。载于第一届软件测试形式化方法与运行时间验证联合国际会议论文集，FATES'06/RV'06，第178–192页，柏林，海德堡，2006年。Springer-Verlag。ISBN 3540496998。doi: 10.1007/11940197 12。URLhttps://doi.org/10.1007/11940197_12。

Zhangyin Feng、Daya Guo、Duyu Tang、Nan Duan、Xiaocheng Feng、Ming Gong、Linjun Shou、Bing Qin、Ting Liu、Daxin Jiang 和 Ming Zhou。CodeBERT：一种预训练模型

用于编程语言和自然语言，2020年。URL https://arxiv.org/abs/2002. 08155。Tianyu Gao、Xingcheng Yao 和 Danqi Chen。SimCSE：句子嵌入的简单对比学习，2022年。URL https://arxiv.org/abs/2104.08821。Jean-Bastien Grill、Florian Strub、Florent Altch´e、Corentin Tallec、Pierre H. Richemond、Elena Buchatskaya、Carl Doersch、Bernardo Avila Pires、Zhaohan Daniel Guo、Moham- mad Gheshlaghi Azar、Bilal Piot、Koray Kavukcuoglu、R´emi Munos 和 Michal Valko。Bootstrap your own latent：一种自监督学习的新方法。载于第34届神经信息处理系统国际会议论文集，NIPS '20，美国纽约州雷德胡克，2020年。Curran Associates Inc. I SBN 9781713829546。Mohammadhosein Hasanbeig、Daniel Kroening 和 Alessandro Abate。结合时序逻辑的深度强化学习。载于定时系统形式化建模与分析国际会议，2020年。URL https://api.semanticscholar.org/CorpusID: 221305630。Dan Hendrycks 和 Kevin Gimpel。高斯误差线性单元（高斯误差线性单元）。arXiv：学习，2016年。URL https://api.semanticscholar.org/CorpusID:125617073。Geoffrey E. Hinton、Oriol Vinyals 和 Jeffrey Dean。提炼神经网络中的知识。arXiv，abs/1503.02531，2015年。URL https://api.semanticscholar.org/ CorpusID:7200347。Corinne Jones、Vincent Roulet 和 Zaid Harchaoui。卷积网络的基于核的变换，2019年。URL https://arxiv.org/abs/1903.08131。Chungpa Lee、Sehee Lim、Kibok Lee 和 Jy yong Sohn。关于对比学习中嵌入的相似性，2025。URL https://arxiv.org/abs/2506.09781。Xiao Li、Cristian-Ioan Vasile 和 Calin Belta。使用时序逻辑奖励的强化学习，2017年。URL https://arxiv.org/abs/1612.03471。Shipeng Liu、Liang Zhao、Deng feng Chen 和 Zhanping Song。用于图像复杂度表示的对比学习。arXiv，abs/2408.03230，2024年。URL https://api.semanticscholar.org/CorpusID:271720179。Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于学习表征国际会议，2017年。URL https://api.semanticscholar.org/CorpusID:53592270。Jamie Mahowald、Brian Bell、Alex Ho 和 Michael Geyer。蒸馏神经切线核的高效分析。2026年。URL https://api.semanticscholar.org/CorpusID: 285541139。Oded Maler 和 D. Niˇckovi´c。监测连续信号的时序性质。载于FORMATS/FTRTFT，2004年。URL https://api.semanticscholar.org/CorpusID: 15642684。

Wonpyo Park、Dongju Kim、Yan Lu 和 Minsu Cho。关系知识蒸馏。*2019*年 *IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），第 3962–3971 页，2019年。网址：https://api.semanticscholar.org/CorpusID:131765296。

Alec Radford 和 Karthik Narasimhan。通过生成式预训练提升语言理解。2018年。网址：https://api.semanticscholar.org/CorpusID:49313245。

Vasumathi Raman、Alexandre Donz´e、Dorsa Sadigh、Richard M. Murray 和 Sanjit A. Seshia。基于信号时序逻辑规约的反应式综合。载于第 *18* 届国际混合系统：计算与控制会议论文集，HSCC '15，第 239–248 页，美国纽约州纽约，2015年。美国计算机协会。ISBN 9781450334334。doi：10.1145/2728606.2728628。网址：https://doi.org/10.1145/2728606.2728628。

Nils Reimers和Iryna Gurevych。Sentence-BERT：使用孪生BERT网络的句子嵌入。载Kentaro Inui、Jing Jiang、Vincent Ng和Xiaojun Wan主编，《*2019*年自然语言处理实证方法会议暨第*9*届自然语言处理国际联合会议论文集》（*EMNLP-IJCNLP*），第3982–3992页，中国香港，2019年11月。计算语言学协会。doi: 10.18653/v1/D19-1410。URLhttps://aclanthology.org/D19-1410/。

Gaia Saveri和Luca Bortolussi。从数据中检索增强挖掘时序逻辑规约，2024年。URL https://arxiv.org/abs/2405.14355。

Gaia Saveri、Laura Nenzi、Luca Bortolussi和Jan Kˇret´ınsk´y。stl2vec：时序逻辑的语义可解释向量表示，2024年。URL https://arxiv.org/abs/2405.14389。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。《你需要的只是注意力》。载于神经信息处理系统，2017年。网址：https://api.semanticscholar.org/CorpusID:13756489。

Tongzhou Wang 和 Phillip Isola。通过超球面上的对齐和均匀性理解对比表示学习，2022年。网址：https://arxiv.org/abs/2005.10242。

Jure Zbontar、Li Jing、Ishan Misra、Yann LeCun 和 St´ephane Deny。Barlow 双胞胎：通过冗余降低进行自监督学习。*arXiv*，abs/2103.03230，2021。URLhttps://api.semanticscholar.org/CorpusID:232110471。

附录A. 核函数实现与 STL 语义

A.1. 核函数实现

第2.2 节中定义的语义内积涉及对信号空间$\mathcal{T}$ 关于测度 μ_0的积分。实际上，语义空间中的这一距离是通过对一组 n 随机信号 $\{\varepsilon_1, ..., \varepsilon_n\} \sim \mu_0$进行蒙特卡洛采样来近似的：

$$\langle \rho(\varphi_i, \cdot),\, \rho(\varphi_j, \cdot) \rangle_{\mu_0} \approx \frac{1}{n} \sum_{t=1}^{n} \rho(\varphi_i, \varepsilon_t) \cdot \rho(\varphi_j, \varepsilon_t)$$

为确保尺度不变性，原始公式采用鲁棒性特征的余弦相似度，而非直接使用点积。设 $||\cdot||$ 表示由 μ_0诱导的希尔伯特空间中的 L_2-范数；归一化相似度 $k'(\cdot,\cdot)$ 的计算方式如下：

$$k'(\varphi_i, \varphi_j) = \frac{\langle \rho(\varphi_i, \cdot),\, \rho(\varphi_j, \cdot) \rangle_{\mu_0}}{||\rho(\varphi_i, \cdot)|| \cdot ||\rho(\varphi_j, \cdot)||} \approx \frac{\sum_{t=1}^{n} \rho(\varphi_i, \varepsilon_t) \cdot \rho(\varphi_j, \varepsilon_t)}{\sqrt{\sum_{t=1}^{n} \rho(\varphi_i, \varepsilon_t)^2} \cdot \sqrt{\sum_{t=1}^{n} \rho(\varphi_j, \varepsilon_t)^2}}$$

最后，通过指数变换将相似度映射到径向基函数（RBF）空间。这会将公式之间的角距离映射为连续谱 $k(\cdot,\cdot) \in [0, 1]$：

$$k(\varphi_i, \varphi_j) = \exp\left(-\frac{2 - 2 \cdot k'(\varphi_i, \varphi_j)}{2\sigma^2} \right)$$

带宽参数 σ^2 （默认设为 0.2）控制语义流形的分辨率。较低的 σ^2 值会提高核的“锐度”，只有在采样轨迹上表现几乎完全相同的公式才会获得较高的相似度分数。

A.2. STL 的布尔与定量语义

设 $\xi\colon\ [0, T] \to \mathbb{R}^n$ 是一条轨迹， φ 是一个 STL 公式。STL 有两种互补的语义：

布尔语义 如果 $\xi \models_t \varphi$，则公式 φ 在时间 t 由 ξ 满足。对于原子谓词 $\mu = x_i \geq \theta$：

$$\xi \models_t \mu \quad \Leftrightarrow \quad x_i(t) \geq \theta.$$

布尔联结词和时序算子按标准方式解释：

$$\begin{aligned}
\xi \models_t \neg\varphi &\Leftrightarrow \xi \not\models_t \varphi, \\
\xi \models_t \varphi_1 \wedge \varphi_2 &\Leftrightarrow \xi \models_t \varphi_1 \text{ and } \xi \models_t \varphi_2, \\
\xi \models_t \varphi_1 \vee \varphi_2 &\Leftrightarrow \xi \models_t \varphi_1 \text{ or } \xi \models_t \varphi_2, \\
\xi \models_t \mathbf{F}_{[a,b]}\varphi &\Leftrightarrow \exists t' \in [t+a, t+b] : \xi \models_{t'} \varphi, \\
\xi \models_t \mathbf{G}_{[a,b]}\varphi &\Leftrightarrow \forall t' \in [t+a, t+b] : \xi \models_{t'} \varphi.
\end{aligned}$$

定量语义（鲁棒性）鲁棒性 $\rho(\varphi,\ \xi,\ t) \in \mathbb{R}$ 衡量满足程度：

$$\rho(\mu,\xi,t) = x_i(t) - \theta, \quad \rho(\neg\varphi,\xi,t) = -\rho(\varphi,\xi,t),$$

$$\rho(\varphi_1 \wedge \varphi_2,\ \xi,\ t) = \min(\rho(\varphi_1,\ \xi,\ t)\ \rho(\varphi_2,\ \xi,\ t)) \quad \rho(\varphi_1 \vee \varphi_2,\ \xi,\ t) = \max(\rho(\varphi_1,\ \xi,\ t)\ \rho(\varphi_2,\ \xi,\ t)),$$

$$\rho(\mathbf{F}_{[a,b]}\varphi,\xi,t) = \max_{t' \in [t+a,t+b]} \rho(\varphi,\xi,t'), \quad \rho(\mathbf{G}_{[a,b]}\varphi,\xi,t) = \min_{t' \in [t+a,t+b]} \rho(\varphi,\xi,t').$$

我们在 $t = 0$ 处求值时，将 $\rho(\varphi,\ \xi) := \rho(\varphi,\ \xi,\ 0)$ 记作。鲁棒性为正表示满足，为负表示违反，其绝对值表示与满足边界的距离。

附录 B. 数据集构建

为确保训练数据集和测试数据集能够有效考验编码器对递归结构的理解能力，我们采用了一种基于对象的增强策略，直接作用于信号时序逻辑（STL）抽象语法树（AST）。

保语义递归增强流水线 该策略的核心是一种作用于 AST 节点的随机优先级级联。对于每个节点，根据概率分布 $P(\cdot)$ 选择一种变换，详见表 3。为确保结构复杂度，该流水线实现了一个强制循环：如果生成的变体未达到目标最小深度（设为 $d \geq 5$），则会重新执行增强，直至满足复杂度阈值。

表 3：保持语义的增强规则。符号说明：$\square$（始终），$\Diamond$（最终），$\mathcal{U}$（直到）。

Rule	$P(\cdot)$	**数学变换**
否定注入	0.1%	$\varphi \to \neg\neg\varphi$ （如果父节点为 $\neq \neg$）
德摩根改写	9.9%	$(A \wedge B) \to \neg(\neg A \vee \neg B)$（并且对偶）
时间划分	35.0%	操作$_I\varphi \to$ 操作$_{I_1}$(操作$_{I_2}\varphi$)，操作 $\in \{\square, \Diamond\}$
直到嵌套	25.0%	$\varphi \to (\varphi\,\mathcal{U}\,\psi)$ 或 $(\psi\,\mathcal{U}\,\varphi)$
时序恒等式	5.0%	$\varphi \to$op$^{[0,0]}\varphi$，op $\in \{\square, \Diamond\}$
分配性	15.0%	$\square_I(A \wedge B) \to \square_I A \wedge \square_I B$ （或对偶）
谓词反转	8.0%	$x_i \leq \theta \to \neg(x_i > \theta)$
无变化（默认）	2.0%	$\varphi \to \varphi$

语义改变扰动 该流程通过**分层方法**生成变体。虽然有些变体仅经历结构复杂化，但数据集中有 65% 会受到语义偏移的影响，从而移动满足边界。这些偏移与结构变换正交应用：公式可以先通过逻辑上的复杂化

达到目标深度，随后再进行数值扰动。扰动在以下情况之间均分（$P = 0.5$）：

- **振动：** 影响阈值（$\pm 10\%$）和时间界限（$W' \in [0.6 \cdot W,\ 1.8 \cdot W]$）的乘性噪声。
- **偏移：** 加性偏移（$\Delta \in [-6, 6]$ 表示阈值，$\Delta \in [-15, 40]$ 表示时间界限）。

这种分层确保语义上不同的公式仍保留较高的句法复杂度，以对编码器进行压力测试。

序列化后精化 序列化完成后，执行三项最终操作，以消除表层句法与逻辑之间的相关性，并确保公式完整性：

- **对偶变换（概率为 40%）：** 通过基于正则表达式的替换，将时序算子替换为其取反后的对偶形式：$\Box_{[I]} A \iff \neg \Diamond_{[I]} \neg A$。
- **区间一致性：** 该算法严格执行时间约束 $R > L$。在进行任何扰动后，上界都会重置为 $R' = L' + W'$，其中新的宽度 W' 以随机方式采样。
- **经验验证：** 每个公式变体都会针对一个随机虚拟信号进行评估。这可确保公式与信号的时间范围兼容，并且不会因嵌套过深或参数偏移而产生未定义的鲁棒性值或空轨迹。

附录 C. 训练指标详解

在我们的设置中，**核对齐** 量化了由 STL 核矩阵 $K \in \mathbb{R}^{B \times B}$ 诱导的语义相似性与从学习得到的嵌入$S \in \mathbb{R}^{B \times B}$ 中获得的神经相似度矩阵之间的全局一致性。

记 vec(·) 为展平算子，我们定义：

$$\text{kernel_alignment}(K, S) = \frac{\text{vec}(K) \cdot \text{vec}(S)}{\|\text{vec}(K)\|_2 \cdot \|\text{vec}(S)\|_2}.$$

该量对应于两个向量化格拉姆矩阵之间的余弦相似度。对齐值为 1。0 表明，学习得到的嵌入保留了所有成对内积（允许存在全局缩放因子），这意味着神经表示近似于 STL 语义希尔伯特空间的几何结构[6]。

均匀性 衡量表示在单位超球面上的分布情况，并可防止表示坍缩。即使成对相似度在局部得到保留，嵌入仍可能因大多数公式集中在空间的某个受限区域而发生退化，从而降低表示的表达能力。

6. 尽管Wang and Isola （2022年）将对齐定义在离散正样本对上，我们的表述将该概念推广到连续相似性结构，以反映由 STL 鲁棒性所诱导的分级语义一致性概念。

根据 Wangand Isola (2022年)，均匀性定义为批次中所有样本对的平均高斯势的对数:

$$\mathcal{U}(Z) = \log\left(\mathbb{E}_{i,j\in B}\left[\exp(-2\|z_i - z_j\|_2^2)\right]\right).$$

这一表述对应于以下情况: $t = 2$ 见 Wang and Isola (2022年)。较高的均匀性表明嵌入广泛分布于超球面上，确保语义不同的规约映射到潜在空间中可分离的区域，从而防止多对一语义坍缩。其取值范围为极端值 $[-4, 0]$，可简要解读如下:

$\mathcal{U}(Z)$	**从几何角度**	**解读**
0.0	完全坍缩	判别能力为零；所有公式均同构
-1.0	密集聚类	分辨率差；不同逻辑占据重叠区域
-2.5	合理分布	标准时序性质的可靠分离
-3.0	良好的稀疏性	最大熵；保留细粒度的语义细微差别。
-4.0	渐近极限	$d \to \infty$ 和 $t = 2$的理论最大分散度

上界意味着完全的维度坍缩，因为只有当所有嵌入的欧几里得距离都为零时才会出现这种情况，即

$$||z_i - z_j||_2^2 = 0, \quad \forall i, j \in B$$

这会使均匀性为零:

$$\mathcal{U}(Z) = \log\left(\mathbb{E}_{i,j\in B}\left[\exp(-2 \cdot 0^2)\right]\right) = \log\left(\mathbb{E}_{i,j\in B}\left[\exp(0)\right]\right) = \log(1) = 0$$

因此，只有当模型无法从句法或语义层面区分不同公式时，我们才会观察到均匀性为零。

相反，理论**下界**对应于超球面上点的完全均匀分布。形式上，对于根据表面积测度 σ_d 均匀分布的无限多个公式，均匀性为:

$$\mathcal{U}(Z) = \log \int_{\mathcal{S}^{d-1}} \int_{\mathcal{S}^{d-1}} \exp\left(-2 \cdot ||u - v||_2^2\right) d\sigma_d(u) d\sigma_d(v)$$

其中 $||u - v||_2^2 = ||u||_2^2 + ||v||_2^2 - 2u^T v = 2 - 2u^T v$ 由于 $u,\ v \in \mathcal{S}^{d-1}$。因此，该表达式可简化为:

$$\mathcal{U}(Z) = -2 \cdot 2 + \log \mathbb{E}_{u,v\sim\sigma_d}\left[\exp\left(2 \cdot 2u^T v\right)\right]$$

由于维度 $d \to \infty$, $u^T v \to 0$，因此（经验）下界为:

$$\mathcal{U}(Z) = -4 + \log \mathbb{E}_{u,v\sim\sigma_d}\left[\exp(0)\right] = -4 + \log(1) = -4$$

附录 D. 时间和内存细节

表格4 和5 中报告的基准测试旨在比较 STL 核与基于 Transformer 的模型。两种方法均在相同的 STL

公式和合成信号数据集上进行评估，以确保结果具有可比性。内存使用量涵盖 CPU 内存和 GPU 显存，并记录嵌入和相似度计算阶段的峰值消耗。

公式的选择是确定性的：对于每次基准测试运行，取数据集中的前 B 个元素用于计算。此阶段不进行随机采样，从而确保各次运行的可复现性。随后，每个公式都会通过基于Transformer的嵌入流程或 STL 核方法进行处理。对于 $Transformer$ 流程，使用其分词器对每个公式进行分词，然后输入训练好的模型以提取嵌入。公式以批次形式通过多线程数据加载器进行分词，并通过前向传播提取嵌入，不进行梯度计算。相似度矩阵通过嵌入的归一化内积计算得出。分别监控随机存取存储器和显存的内存使用量，并使用 GPU 同步计时器测量时间，以确保精确测量前向传播和相似度计算所用的时间。基准测试在两种场景下重复进行：一种是模型已加载，另一种则包含完整的模型加载过程。对于*STL* 核 方法，首先对合成信号进行采样，生成 N 条轨迹，每条轨迹包含 1000 个点和 3 个变量。随后，在这些信号上评估公式，以计算其鲁棒性向量。最后，将核矩阵计算为鲁棒性向量之间的相似度测度。记录随机存取存储器（RAM）和显存（VRAM）的内存使用量及耗时，涵盖鲁棒性计算和核评估期间的峰值消耗。

所有实验均在相同的硬件配置（NVIDIA A100 80GB GPU）上运行，以确保对两种方法进行公平比较。内存使用量包括 CPU 内存和 GPU 显存分配，通过系统级查询和 PyTorch 内存跟踪进行测量。计时涵盖所有 GPU 同步点，以避免低估实际运行时间。

表4：STL 核和 Transformer 的嵌入时间（秒）及总内存使用量（MB）。总内存 = 随机存取存储器 + 显存。“已加载”指模型已初始化的 Transformer；“完整流程”包括模型加载。

B	N	STL 核		Transformer			
		T_{Emb}	Mem	T_{Emb}（已加载）	内存（已加载）	T_{Emb}（完整流程）	内存（完整）
500	500	0.61	2059	0.62	2136	2.06	2914
500	1000	0.83	3061	0.62	2128	1.98	2916
500	2000	1.25	5051	0.62	2128	2.05	2918
500	4000	2.21	8955	0.63	2134	2.02	2918
500	8000	4.53	16995	0.62	2127	1.98	2918
500	16000	12.44	32442	0.62	2126	1.99	2918
1000	500	1.09	3027	1.11	2139	2.53	2911
1000	1000	1.54	4960	1.11	2138	2.54	2911
1000	2000	2.40	8829	1.11	2138	2.53	2910
1000	4000	4.63	16505	1.11	2139	2.48	2910
1000	8000	9.10	32008	1.11	2139	2.47	2909
1000	16000	24.70	62338	1.11	2138	2.56	2910
2000	500	2.18	4907	2.17	2158	3.59	2914
2000	1000	3.03	8737	2.16	2151	3.56	2914
2000	2000	4.77	16464	2.18	2155	3.59	2914
2000	4000	8.36	31696	2.15	2156	3.52	2915
2000	8000	17.08	62384	2.16	2152	3.55	2912
2000	16000	48.86	123384	2.17	2143	3.56	2913
4000	500	4.33	8755	4.45	2163	5.91	2992
4000	1000	6.09	16452	4.41	2162	5.84	2988
4000	2000	9.47	31897	4.42	2163	5.88	2985
4000	4000	17.06	62408	4.38	2162	5.86	2986
4000	8000	34.30	123788	4.42	2158	5.84	2980
4000	16000	232.63	246151	4.46	2157	6.20	2980

核方法和基于Transformer的嵌入流水线的计算扩展性。运行时间（图D）和内存使用量（图 D）以信号分辨率 N 和批量大小 B为变量进行报告。预测值（虚线）表示超出内存限制的配置（内存溢出）。

表 5：STL 核和 Transformer 的成对相似度计算时间（秒）及总内存使用量（MB）。总内存 = 随机存取存储器 + 显存。“已加载”表示模型已初始化的 Transformer；“完整流程”包括模型加载。对于较大的 N，核会发生内存溢出。

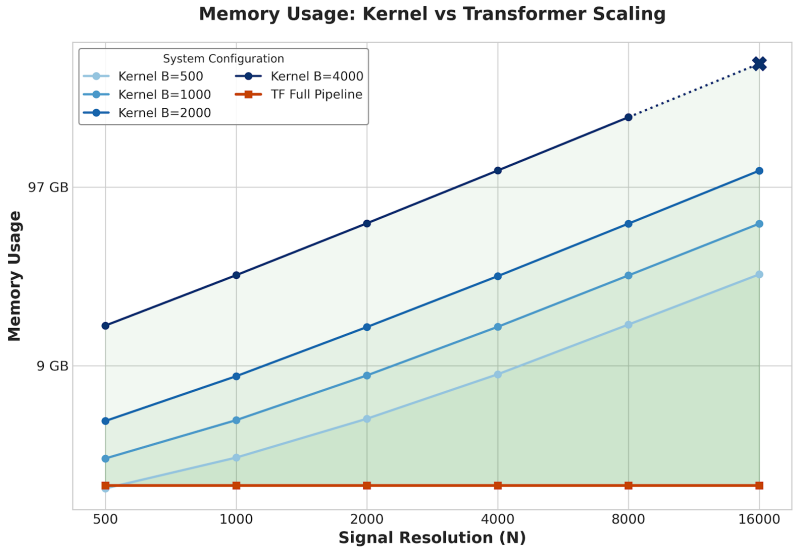
B	N	STL 核		Transformer（已加载）		Transformer（完整流程）	
		T_{Sim}	Mem	T_{Sim}	Mem	T_{Sim}	Mem
500	500	1.58	2059	0.73	2136	2.04	2914
500	1000	2.50	3061	0.73	2128	1.99	2916
500	2000	4.38	5051	0.73	2128	2.02	2918
500	4000	8.21	8955	0.73	2134	2.02	2918
500	8000	16.81	16995	0.73	2127	1.99	2918
500	16000	40.24	32442	0.73	2126	1.99	2918
1000	500	3.64	3027	1.21	2139	2.52	2911
1000	1000	6.17	4960	1.22	2138	2.53	2911
1000	2000	11.05	8829	1.21	2138	2.52	2910
1000	4000	21.57	16505	1.22	2139	2.47	2910
1000	8000	42.86	32008	1.22	2139	2.45	2909
1000	16000	98.51	62338	1.21	2138	2.56	2910
2000	500	10.54	4907	2.29	2158	3.53	2914
2000	1000	18.33	8737	2.28	2151	3.59	2914
2000	2000	33.87	16464	2.29	2155	3.62	2914
2000	4000	66.37	31696	2.27	2156	3.53	2915
2000	8000	132.29	62384	2.28	2152	3.54	2912
2000	16000	434.57	123384	2.29	2143	3.55	2913
4000	500	32.65	16776	4.59	2163	5.85	2992
4000	1000	60.11	32112	4.55	2162	5.93	2988
4000	2000	114.98	62653	4.53	2155	6.05	2985
4000	4000	227.25	123651	4.51	2156	6.47	2986
4000	8000	552.04	245724	4.58	2152	5.83	2980
4000	16000	1089.28*	489898*	4.65	2152	6.01	2980

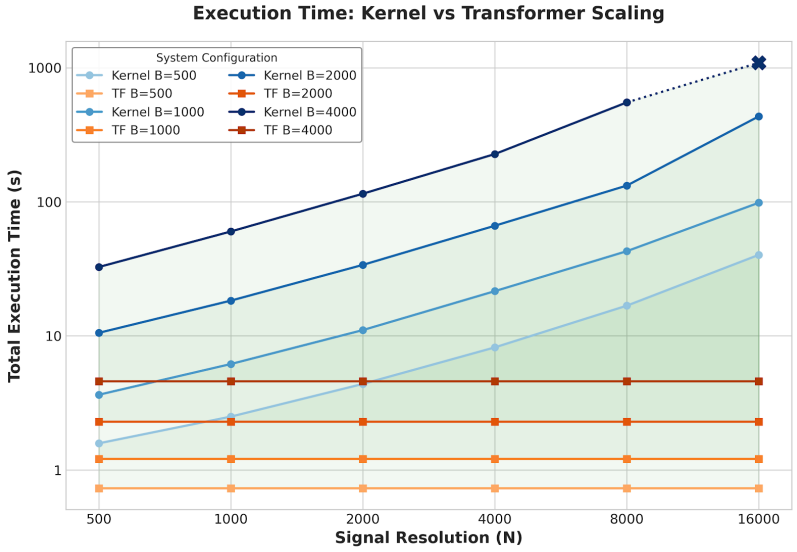
附录 E. [CLS]、[BOS] 与均值池化比较

在第4节中，我们介绍了三种不同的池化策略，用于将 Transformer 编码器产生的词元级表示聚合为单一固定维度的嵌入。为评估我们的蒸馏框架对所选聚合方法的敏感性，我们针对每种策略给出了详细的语义理解评估。

表1 （为便于比较，此处从正文转载）、表7和表8 分别报告了 [CLS],均值以及 [BOS] 池化的性能。在所有度量上，结果表明所提出的核对齐目标对池化操作的选择具有高度稳健性。

尽管 [CLS] 池化与目标核相似度的对齐略微更紧密（在等价对上取得最低的平均绝对误差0.034），两者





均值池化与［BOS］池化成功内化了连续语义几何。这三种策略均能持续且有效地将语义等价公式与不等价公式及语法相似公式区分开来。这证实了该架构瓶颈能够有效压缩逻辑含义并滤除语法噪声，而与所采用的具体词元聚合启发式方法无关。

表 6： [CLS]池化的语义理解评估（参考）。

度量	等价	不等价	句法相似
神经相似度	0.966	0.182	0.308
核相似度	0.997	0.170	0.225
平均绝对误差（语义差距）	0.034	0.072	0.112
相对神经距离	0.137	0.833	0.770
相对核距离	0.105	0.830	0.775

表 7：均值池化的语义理解评估。

度量	等价	不等价	句法相似
神经相似度	0.954	0.201	0.332
核相似度	0.997	0.163	0.221
平均绝对误差（语义差距）	0.044	0.066	0.143
相对神经距离	0.169	0.833	0.749
相对核距离	0.100	0.837	0.779

表8： [BOS]池化的语义理解评估。

度量	等价	不等价	句法相似
神经相似度	0.946	0.171	0.280
核相似度	0.997	0.152	0.182
平均绝对误差（语义差距）	0.052	0.070	0.133
相对神经距离	0.186	0.834	0.794
相对核距离	0.111	0.848	0.818