arXiv:2603.07415v1 [cs.LG] 8 Mar 2026

上下文信道容量：理解灾难性遗忘的信息论框架

Ran Cheng

2026年3月10日

摘要

灾难性遗忘仍是持续学习（CL）中的核心挑战，但目前尚缺乏统一的信息论解释，说明为何有些架构会发生灾难性遗忘，而另一些则不会。我们提出上下文信道容量 (C_{ctx})，即持续学习架构的上下文信号与其生成参数之间的互信息，并证明零遗忘要求 $C_{\text{ctx}} \geq H(T)$，其中$H(T)$是任务身份熵。我们确立了不可能三角——基于状态的顺序学习器无法同时满足零遗忘、在线学习和有限参数——并表明，条件再生成架构（超网络）通过将参数重新定义为函数值而非状态，绕开了这一三角关系。

我们在 Split-MNIST 上对 8 种持续学习方法进行了验证（86 天内开展 1,130+ 次实验，每种方法使用 4个随机种子），结果显示，C_{ctx} 能够完美预测遗忘行为：具有 $C_{\text{ctx}} = 0$的 方法（NaiveSGD、EWC、突触智能、LwF、CFlow）会发生灾难性遗忘（6–97%），而具有 $C_{\text{ctx}} \approx 1$的 方法（超网络）则实现了零遗忘（准确率为 98.8%）。我们还提出错误上下文探测 (P5)，这是一种用于测量 C_{ctx}的实用诊断方案，并通过一种新颖的梯度上下文编码器 将该框架扩展到 CIFAR-10，将预言机差距从 23.3 个百分点缩小至 0.7 个百分点。我们系统梳理了一系列 15+已关闭的研究方向，包括赫布零结果（冻结随机特征优于学习特征）、CFlow 的 θ_0-记忆器现象，以及阻碍列专门化的 S_N 对称性障碍，为学界提供了经过精确诊断的负面结果。我们的核心设计原则是：架构优先于算法——上下文通路必须在结构上无法绕过。

1 引言

灾难性遗忘——神经网络按顺序学习新任务时，突然丧失先前习得的知识——早在三十多年前便有文献记载（McCloskey和Cohen，1989），至今仍是机器学习领域一个尚未解决的核心问题（French，1999）。尽管研究者持续投入精力，该领域仍缺乏一个有原则、可量化的框架，来解释为何有些持续学习（CL）架构会发生灾难性遗忘，而另一些则不会。

现有的遗忘缓解方法大致分为三类。基于正则化的方法，例如弹性权重巩固（EWC）（Kirkpatrick等人，2017）和突触智能（SI）（Zenke等人，2017），会惩罚被认为对过去任务十分重要的参数变化。基于重放的方法会存储历史任务的样本示例，并在训练过程中交替使用，从而有效打破严格的顺序约束。基于架构的方法，包括渐进式网络（Rusu等人，2016）和超网络（von Oswald 等人，2020；Ehret 等人，2021），会分配或生成任务特定参数。每一类方法都催生了效果各异的方法，但其性能表现差异显著：在 Split-MNIST 上，EWC 的准确率为18.9%，而超网络的准确率达到98.8%——两者相差*80*个百分点，而所用架构的参数数量相当（表1）。

缺失的解释。是什么造成了这80个百分点的差距？这并非仅仅是“算法更复杂”的问题：EWC 使用费舍尔信息，SI 跟踪路径积分，而 CFlow 求解神经常微分方程——这些方法都可以说比 HyperNetwork 使用的简单多层感知机更复杂。也不是容量问题：CFlow 系统有 270 万个参数，而 HyperNetwork 的参数不到 10 万个。此前对持续学习进行的信息论分析（Achille 和 Soatto，2018；Doan 等人，2021；Taheri 和 Thrampoulidis，

表1：8种持续学习方法在 Split-MNIST 上的结果（每种方法使用4个随机种子）。准确率：完成全部5个任务后的平均准确率。遗忘：平均遗忘。P5 Δ：错误上下文探测（§??）下的准确率下降。ˆ C_{ctx}：实证上下文信道容量代理指标。方法按其有效的 C_{ctx} 机制划分。表中呈现出鲜明的二分现象：所有具有 $C_{\mathrm{ctx}} = 0$ 的方法都表现出显著遗忘，而具有 $C_{\mathrm{ctx}} \gg 0$ 的方法则实现了零遗忘。

范式	方法	准确率（%）↑	遗忘率（%）↓	P5 Δ	$\hat{C}_{\mathrm{ctx}}$
状态保护 ($C_{\mathrm{ctx}} = 0$)	NaiveSGD	18.7 ± 0.3	97.1 ± 0.4	0.0	0.000
	EWC	18.9 ± 0.1	97.6 ± 0.6	0.0	0.000
	SI	16.4 ± 3.7	97.1 ± 0.4	0.0	0.000
	LwF	24.2 ± 0.6	54.8 ± 5.5	0.0	0.000
重放 ($C_{\mathrm{ctx}} = 0$)	经验回放	85.9 ± 1.3	12.5 ± 1.7	0.0	0.000
状态变换。 ($C_{\mathrm{ctx}} \approx 0$)	CFlow（常微分方程）	92.4 ± 0.7	6.1 ± 1.1	0.0	0.000
条件再生 ($C_{\mathrm{ctx}} \gg 0$)	HyperNet 预言机	$\mathbf{98.8 \pm 0.3}$	$\mathbf{0.0 \pm 0.0}$	-97.6	**0.976**
	HyperNet 学习式	98.9 ± 0.2	0.0 ± 0.0	-95.2	0.952

2025）已经针对特定情境给出了遗忘率的上界，但尚未回答根本的架构问题：架构的哪种结构属性决定了遗忘是不可避免还是可以避免？

我们的答案：上下文信道容量。我们提出，决定遗忘的关键量是**上下文信道容量**（C_{ctx}）：持续学习架构的上下文信号与其用于预测的参数之间的最大互信息。我们的核心理论结果（定理4）指出，任何在 K 个任务上运行的持续学习架构 $\mathcal{A}$，其期望遗忘满足

$$\overline{\mathrm{Fgt}}(\mathcal{A}, K) \;\geq\; \max\!\left(0,\; 1 - \frac{C_{\mathrm{ctx}}(\mathcal{A})}{H(T)}\right) \cdot \overline{\mathrm{Fgt}}_{\max}, \tag{1}$$

其中，$H(T) = \log_2 K$ 表示任务身份熵。对于没有上下文通路的架构（EWC、突触智能、NaiveSGD），$C_{\mathrm{ctx}} = 0$，无论学习算法如何，遗忘程度都达到最大。对于超网络，$C_{\mathrm{ctx}} \gg H(T)$，且可以实现零遗忘。该上界是紧的：它完美地区分了表1中的8种方法。

贡献。我们的贡献有四项：

1. **上下文信道容量**（C_{ctx}），一种新的信息论量，可统一分析各种持续学习架构。我们证明了不可能三角（定理 3）——在序列状态更新条件下，零遗忘、在线学习和有限参数三者无法兼得——以及*CCC*界（定理4），该界给出了遗忘量关于 C_{ctx}的下界。

2. 一种将持续学习方法分为三类的**范式分类**——状态保护（$C_{\mathrm{ctx}} = 0$）、状态变换（$C_{\mathrm{ctx}} \to 0$，由于结构性绕过），以及条件再生（$C_{\mathrm{ctx}} \gg H(T)$）——并证明只有第三类能够在有界参数下实现零遗忘。

3. **错误上下文探测**（P5），一种实用的实证实验方案，为 C_{ctx}提供了经验代理指标：使用故意错误的上下文评估上下文条件模型，并测量准确率下降幅度。较大的 P5 变化量表示较高的 C_{ctx}；变化量为零则表示上下文被绕过。

4. **全面的实证验证**覆盖 Split-MNIST 上的 8 种持续学习方法（1,130+ 次实验，历时 86 天，每种配置 4 个随机种子），表明 C_{ctx} 能够完美预测所测试的每种方法的遗忘区间。

更广泛的贡献：系统性负面结果。历经 86 天以及 $15+$ 个研究方向，我们积累了大量负面结果——每一项都经过系统记录、可复现且富有信息量。我们表明，当组合容量充足时，赫布学习对持续学习性能的贡献恰好为零（§2.5.1）；当上下文是拼接而非条件化时，基于常微分方程的参数流会退化为 θ_0记忆器（§2.5.2）；并且列专门化无法在齐次架构中从局部学习规则中涌现。这些负面结果若透过 C_{ctx}的视角来看，便成为一段连贯的叙述，而非一连串失败。

一句话总结。 架构决定命运：一个持续学习系统是否遗忘，并不由其学习算法决定，而取决于其架构是否提供了一条不可绕过且具有足够容量的上下文通路——我们将这一原则形式化为 $C_{\mathrm{ctx}} \geq H(T)$。

2 理论框架

我们分五个阶段构建持续学习的信息论框架。首先，我们将持续学习形式化为受限在线编码（§2.1）。其次，我们证明对于基于顺序状态的学习器而言，遗忘在信息论上是不可避免的（§2.2）。第三，我们证明一个不可能三角（§2.3）。第四，我们引入上下文信道容量并推导CCC界（§2.4）。第五，我们按其 C_{ctx} 机制对持续学习架构进行分类，并解释超网络为何能够绕过不可能三角（§2.5–§2.6）。

2.1 持续学习作为受限在线编码

定义1 （持续学习问题）. 一个持续学习器是一个元组$(\Theta, F, U, \{D_k\}_{k=1}^{K})$其中 *e:*

- $\Theta \subseteq \mathbb{R}^d$ 是参数空间，其中 $\dim(\Theta) = d$；ˆ
- F： $\mathcal{X} \times \Theta \to \mathcal{Y}$ 是预测函数；
- U： $\Theta \times \mathcal{D} \to \Theta$ 是更新规则（例如随机梯度下降、*Hebbian*）；
- $\{D_k\}_{k=1}^{K}$ 是依次到达的任务数据集，其中每个 D_k 均独立同分布地采样自特定任务分布 $P_k(X, Y)$。

学习过程会产生参数轨迹 $\theta_0 \to \theta_1 \to \cdots \to \theta_K$，其中 $\theta_k = U(\theta_{k-1}, D_k)$。持续学习的定义性约束是因果性：处理任务 D_k时，学习者只能访问 θ_{k-1} 和D_k，而无法访问先前任务的原始数据 $D_1, \ldots, D_{k-1}$。[1]

定义 2（遗忘）。 学习任务 $k > j$ 后，任务 j 的遗忘 为

$$\mathrm{Fgt}_j^k \;=\; \max_{j \leq t \leq k-1} \mathrm{ACC}_j(\theta_t) \;-\; \mathrm{ACC}_j(\theta_k), \tag{2}$$

其中，准确率$_j(\theta)$ 表示参数 θ 在任务 j的测试集上的准确率。经过K 个任务后的平均遗忘 为

$$\overline{\mathrm{Fgt}} \;=\; \frac{1}{K-1} \sum_{j=1}^{K-1} \mathrm{Fgt}_j^K. \tag{3}$$

与率失真理论的联系。 持续学习问题可以视为一个受约束的在线信源编码问题。参数空间 Θ是容量为 $C \approx d \cdot \log_2(1/\delta)$ 比特的信道，其中 δ是有效参数精度。每个任务 D_k 都是一个必须编码到该信道中的“信源”，而失真就是分类误差。与标准率失真理论的关键区别在于因果约束：时刻 k 的编码器只能看到当前信源 D_k 和前一个码字 θ_{k-1}，而无法看到完整的信源序列（D_1，……， D_K）。这使得持续学习严格难于联合（离线）编码，后者可以达到率失真最优。

[1]~~基于回放的方法通过维护缓冲区 $\mathcal{B} \subset \bigcup_{j<k} D_j$~~，部分违反了这一约束。我们在 §2.5.1 中明确分析了这种放宽。

定义 3（参数容量）。 对于有效单参数精度为$\delta > 0$的参数空间 $\Theta \subseteq \mathbb{R}^d$，参数容量 为

$$C(\Theta) \;=\; d \cdot \log_2(1/\delta) \quad bits. \tag{4}$$

对于 *32* 位浮点参数， $C \approx 32d$。在实践中，由于损失景观中存在冗余，有效容量要低得多。

2.2 信息瓶颈链

现在我们将证明，对于任何容量有限的序列学习器而言，遗忘在信息论意义上不可避免。

定理1 (遗忘在信息论意义上不可避免**)**。 对于参数容量为 $C = d \cdot \log_2(1/\delta)$ 比特的确定性序列学习器，以及相互独立的任务 $\{D_k\}_{k=1}^{K}$，最终参数与任意过去任务数据集之间的互信息满足：

$$I(\theta_K; D_1) \;\le\; I(\theta_{K-1}; D_1) \;\le\; \cdots \;\le\; I(\theta_1; D_1) \;\le\; C. \tag{5}$$

证明。 由于 $\theta_k = U(\theta_{k-1},\ D_k)$ 是一个确定性函数，且给定 θ_{k-1} 时 D_k 与 D_1 相互独立（这是由因果约束和任务独立性所致），这些随机变量构成一个马尔可夫链：

$$D_1 \;\to\; \theta_1 \;\to\; \theta_2 \;\to\; \cdots \;\to\; \theta_K. \tag{6}$$

验证马尔可夫性质：给定 θ_{k-1}，变量 $\theta_k = U(\theta_{k-1},\ D_k)$ 仅通过 θ_{k-1} 依赖于 D_1，因为 D_k 与所有先前的数据相互独立。根据数据处理不等式（DPI），对于任意马尔可夫链$X \to Y \to Z$，我们有 $I(X; Z) \le I(X; Y)$。迭代应用可得：

$$I(\theta_K; D_1) \;\le\; I(\theta_{K-1}; D_1) \;\le\; \cdots \;\le\; I(\theta_1; D_1). \tag{7}$$

初始上界源于这样一个事实： $\theta_1 \in \Theta$ 对于任何量，最多只能表示 C 比特的信息，因此 $I(\theta_1; D_1) \le H(\theta_1) \le C$。 □

推论 2（遗忘下界）。 如果 K 个任务相互独立，且每个任务都至少需要$R_{\min}$ 比特的任务特定信息才能达到非平凡的性能（即， I（θ； D_k） $\ge R_{\min}$ 是准确率$_k$（θ） $>$ 准确率$_{\text{chance}}$所必需的），那么当 $K > C/R_{\min}$时，不可能同时在所有 K个任务上保持非平凡的性能。

证明。 根据马尔可夫链，所有任务特定信息都必须经过瓶颈 θ_K。由于这些任务相互独立，要在所有 K 个任务上达到非平凡性能，所需的总信息量至少为$K \cdot R_{\min}$。但是， $I(\theta_K; D_1, \ldots\ldots, D_K) \le H(\theta_K) \le C$。当 $K \cdot R_{\min} > C$时，鸽巢原理意味着至少有一个任务满足 $I(\theta_K; D_j) < R_{\min}$，因此 ACC$_j(\theta_K) \le$ ACC$_{\text{chance}}$。 □

与先前工作的关系。 定理1将这一观点形式化：遗忘是有限容量的必然结果。French（1999）曾非正式地提出这一观点，而 Taheri 和 Thrampoulidis（2025）则针对单隐层网络对其进行了部分形式化。我们的贡献在于将其表述为对马尔可夫链应用数据处理不等式（DPI）的结果，这种表述（i）不需要对网络架构或训练过程作任何假设；（ii）明确指出了因果约束 的作用：正是序贯结构 $D_1 \to \theta_1 \to \cdots \to \theta_K$ 造成了信息瓶颈。打破这一链条的方法（例如重放、联合训练）不受公式（5）的约束。

2.3 不可能三角

定理1表明，关于过去任务的信息会在序贯更新过程中单调丢失。现在，我们将这一结论进一步强化为一个三难困境。

定理 3 (持续学习不可能三角**)**。 基于状态的顺序学习器无法同时满足以下三种性质：

表 2：现有持续学习方法如何"逃离"不可能三角。每种方法都会放宽三种性质之一。HyperNet 通过将 θ 重新定义为函数值而非状态 (§2.6)，成为唯一满足全部三种性质的方法。

方法	Zero 遗忘	在线 学习	有界 $\lvert\theta\rvert$	逃逸机制
EWC / 突触智能 / LwF	$\approx$	✓	✓	近似零遗忘
经验回放	✓	×	✓	违反因果约束（重新访问数据）
渐进式网络	✓	✓	×	参数随 K 线性增长
HyperNet	✓	✓	✓*	$\theta_k = g(c_k)$：参数是函数值

*元学习参数（V，g，$\theta_{\text{基础}}$）在联合训练后固定；θ_k 按每个任务重新生成。

1. 零遗忘：准确率$_j(\theta_k)$ =准确率$_j(\theta_j)$ 对所有 $j < k \leq K$。

2. 在线学习（因果约束）：θ_k 仅依赖于 θ_{k-1} 和 D_k。

3. 有界参数：$|\theta| = d$ 不随 K增长。

证明。假设这三种性质均成立。我们将推出一个矛盾。

步骤 *1*：信息累积。性质 (1) 要求，对于每个 $j \in \{1, \ldots, K\}$，最终参数θ_K 在任务 j 上的准确率必须与 θ_j 在刚完成针对 D_j的训练后所达到的准确率相同。这意味着 θ_K为每项任务保留了足够的任务特定信息：

$$I(\theta_K; D_j) \;\geq\; R_j \quad \text{for all } j \in \{1, \ldots, K\}, \tag{8}$$

wh其中 $R_j > 0$ 是达到准确率 ACC$_j(\theta_j)$ 所需的最小互信息 .

步骤 *2*：独立 性意味着可加性。由于任务 $\{D_k\}$是独立抽取的，总 信息需求 量是

$$I(\theta_K; D_1, \ldots, D_K) \;\geq\; \sum_{j=1}^{K} R_j \;\geq\; K \cdot R_{\min}, \tag{9}$$

其中 $R_{\min} = \min_j R_j > 0$。

步骤 *3*：容量界。性质 (2) 和 (3) 共同表明，$\theta_K \in \Theta$ 是由因果过程生成的固定维度向量。因此，$H(\theta_K) \leq C = d \cdot \log_2(1/\delta)$，其值在 K 中为常数。

步骤 *4*：矛盾。对于 $K > C/R_{\min}$，我们有 $K \cdot R_{\min} > C \geq H(\theta_K) \geq I(\theta_K; D_1, \ldots, D_K)$，这与步骤 2 矛盾。因此，这三种性质中至少有一种必须被违反。 □

表 2 展示了 每类持续学习方法如何通过放宽不同的性质来化解这一三难困境 y.

注 1（不可能三角的尖锐性）。这种不可能性并非仅存在于渐近情形。在 *Split-MNIST*（$K = 5$）上，*EWC* 仅达到 *18.9%* 的准确率——与 *NaiveSGD* 的 *18.7%* 基本相当——这表明，当参数容量 C 已被顺序更新过程耗尽时，费舍尔信息正则化几乎无法提供保护。*EWC* 中的正则化项控制着哪些信息会被覆盖，但无法增加总容量 C。

2.4 上下文通道容量

不可能三角（定理 3）适用于基于状态的顺序学习器——即 θ_k 通过修改 θ_{k-1}而获得的系统。现在，我们引入区分能够绕过这一限制的架构的关键概念。

定义 4（上下文信号）。对于持续学习架构，上下文信号是任何携带当前任务身份信息的输入 $c \in \mathcal{C}$。示例包括：预言机任务 *ID*（独热向量）、批次统计量（当前输入的平均值、方差）、梯度信号或学习式任务嵌入。对于没有显式上下文机制的架构（例如随机梯度下降、*EWC*、突触智能），我们将 $c = \emptyset$ 定义为（无上下文信号）。

定义 5（上下文通道及其容量）。对于具有上下文信号 $c \in \mathcal{C}$ 和参数生成机制 $\theta(c)$ 的持续学习架构，上下文通道是映射 $c \mapsto \theta(c)$。上下文信道容量为：

$$C_{\text{ctx}} \;=\; \max_{P(c)} I(c;\, \theta(c))\,, \tag{10}$$

其中，最大值取遍上下文空间 $\mathcal{C}$ 上的所有分布。

对于不含显式上下文通路的架构（定义 *4*，$c = \emptyset$），我们定义 $C_{\text{ctx}} = 0$。[2]

直观地说。C_{ctx} 衡量有多少位任务识别信息能够通过架构的上下文通路流动，从而影响预测参数。如果 $C_{\text{ctx}} = 0$，该架构就没有机制为不同任务生成不同的参数，因此必须对所有任务使用同一组参数，从而导致干扰。如果 $C_{\text{ctx}} \geq H(T)$，原则上，该架构能够为每个任务生成完全不同的参数，从而彻底消除干扰。

现在我们陈述主要的理论结果。

定理 4（上下文信道容量界——CCC界）。设 $\mathcal{A}$ 是一个持续学习架构，处理 K 个任务，任务身份随机变量为 $T \in \{1, \ldots, K\}$。令 $H(T) = \log_2 K$（假设任务服从均匀分布）。则期望遗忘满足：

$$\overline{\text{Fgt}}(\mathcal{A}, K) \;\geq\; \max\!\left(0,\; 1 - \frac{C_{\text{ctx}}(\mathcal{A})}{H(T)}\right) \cdot \overline{\text{Fgt}}_{\max}, \tag{11}$$

其中 $\overline{\text{Fgt}}_{\max}$ 是仅保留最近任务的模型所产生的遗忘（对所有先前任务进行随机猜测）。

证明。我们分三步进行。

步骤 *1*：通过上下文进行任务识别。考虑以下通信问题：自然抽取一个任务 $T \sim \text{Uniform}\{1, \ldots, K\}$，并给出数据 D_T。架构生成上下文 c_T（可能是随机的），并生成参数 $\theta_T = g(c_T)$。要在任务 T 上正确预测，生成的参数 θ_T 必须编码关于 T 的充分信息。具体而言，考虑一个解码器 $\hat{T} = \phi(\theta_T)$，尝试根据生成的参数识别任务。正确识别任务的概率受 Fano 不等式上界限制：

$$P(\hat{T} \neq T) \;\geq\; \frac{H(T \mid \theta_T) - 1}{\log_2(K-1)}. \tag{12}$$

步骤 *2*：通过上下文通道对 $H(T \mid \theta_T)$ 进行上界估计。由于 $\theta_T = g(c_T)$ 和 $T \to c_T \to \theta_T$ 构成一个马尔可夫链（上下文由任务数据生成，参数由上下文生成），数据处理不等式给出：

$$I(T; \theta_T) \;\leq\; I(T; c_T) \;\leq\; I(c_T; \theta(c_T)) \;\leq\; C_{\text{ctx}}. \tag{13}$$

第一个不等式是将数据处理不等式应用于 $T \to c_T \to \theta_T$。第二个利用了 T 决定 c_T 这一事实（因此 $I(T;\, c_T) \leq H(c_T)$），以及根据信道容量的定义，$I(c_T;\, \theta(c_T)) \leq C_{\text{ctx}}$。因此：

$$H(T \mid \theta_T) \;=\; H(T) - I(T; \theta_T) \;\geq\; H(T) - C_{\text{ctx}}. \tag{14}$$

步骤*3*：将任务误识别与遗忘联系起来。当架构生成“错误”参数——即适用于任务 $T' \neq T$ 的参数——时，任务 T 上的准确率会下降到大约随机猜测水平。[3] 期望遗忘至少与任务误识别的概率一样大（直至最大遗忘）：

$$\overline{\text{Fgt}} \;\geq\; P(\hat{T} \neq T) \cdot \overline{\text{Fgt}}_{\max}. \tag{15}$$

结合方程式 (12) 以及对 $H(T \mid \theta_T)$ 的上界：

$$\overline{\text{Fgt}} \;\geq\; \frac{H(T) - C_{\text{ctx}} - 1}{\log_2(K-1)} \cdot \overline{\text{Fgt}}_{\max}. \tag{16}$$

[2]这与式(10)一致：如果 θ 不依赖于任何输入 c，则对所有 $P(c)$，$I(c;\, \theta(c)) = 0$。[3]这是一种近似，当各任务不共享类别结构时，该近似变为精确。对于在10个类别上包含5个二元任务的Split-MNIST，错误任务参数会产生约10%的准确率。

对于 $K \geq 3$，$\log_2(K-1) \leq H(T)$，并去掉 -1 项（这会收紧上界），我们得到所述结果：

$$\overline{\mathrm{Fgt}} \;\geq\; \max\!\left(0,\; 1 - \frac{C_{\mathrm{ctx}}}{H(T)}\right) \cdot \overline{\mathrm{Fgt}}_{\max}. \tag{17}$$

□

CCC界有两个直接且重要的结果：

推论 5（$C_{\mathrm{ctx}} = 0$ 意味着最大遗忘）。 如果 $C_{\mathrm{ctx}}(\mathcal{A}) = 0$（无上下文通路），则

$$\overline{\mathrm{Fgt}}(\mathcal{A}, K) \;\geq\; \overline{\mathrm{Fgt}}_{\max} \tag{18}$$

*reg*无论学习算法、正则化强度或任何其他超参数如何。

这一推论具有惊人的实证结果：它预测NaiveSGD、EWC和突触智能应表现出相似的遗忘水平，因为它们都具有 $C_{\mathrm{ctx}} = 0$。表1证实了这一点：三者均达到16–19%的准确率，遗忘率为97%。EWC中的Fisher正则化与突触智能中的路径积分加权并未提供任何保护——它们无法改变该架构没有产生任务特定参数的机制这一事实。

推论6（零遗忘的充分条件）。 如果 $C_{\mathrm{ctx}}(\mathcal{A}) \geq H(T)$ 且映射 $c \mapsto \theta(c)$ 具有足够的表达能力（即，存在一种生成器参数设定，使得 $\theta(c_k)$ 在任务 k 上对每个 k都能达到最优准确率），则零遗忘是可实现的。

证明。 当 $C_{\mathrm{ctx}} \geq H(T) = \log_2 K$ 时，上下文通道能够传输至少 $\log_2 K$ 位信息，足以唯一识别每个 K 任务。只要生成器具有足够的表达能力，就能将每项任务映射到其最优参数配置。由于参数是在推理时根据上下文重新生成的（而非作为状态予以维护），因此不会发生顺序覆写，也就不会产生遗忘。

□

2.5 基于 C_{ctx} 的范式分类

C_{ctx} 框架基于信息流架构，自然地将持续学习方法划分为三种范式。这种分类不仅仅是描述性的——每种范式都具有可证明的不同遗忘特性。

2.5.1 范式 A：状态保护（$C_{\mathrm{ctx}} = 0$）

在这种范式中，参数向量 θ 是一种状态，会被顺序更新：

$$\theta_k \;=\; U(\theta_{k-1}, D_k), \quad \text{subject to a protection constraint } \Omega(\theta_k, \theta_{k-1}). \tag{19}$$

保护约束 Ω 可以是围绕重要参数的二次惩罚（EWC：$\Omega = \frac{\lambda}{2}\sum_i F_i(\theta_k^{(i)} - \theta_{k-1}^{(i)})^2$）、路径积分重要性（突触智能），也可以是知识蒸馏（LwF）。关键在于，不存在上下文信号：所有任务都使用相同的参数 θ_K 进行预测。

定理 7（状态保护存在不可消除的稳定性—可塑性权衡）。 对于任何保护强度为 $\lambda \geq 0$ 的范式 A 方法，都存在如下权衡：

$$I(\theta_K; D_K) + I(\theta_K; D_1, \ldots, D_{K-1}) \;\leq\; C, \tag{20}$$

其中，$I(\theta_K;\, D_K)$ 衡量可塑性（关于当前任务的信息），而 $I(\theta_K;\, D_1, \ldots, D_{K-1})$ 衡量稳定性（保留的关于过去任务的信息）。增加 λ 会将信息从可塑性转向稳定性，但无法增加信息总量。

证明。 由于 $\theta_K \in \Theta$ 的熵 $H(\theta_K) \leq C$有界，且根据互信息链式法则：

$$I(\theta_K; D_1, \ldots, D_K) \;=\; I(\theta_K; D_K) + I(\theta_K; D_1, \ldots, D_{K-1} \mid D_K). \tag{21}$$

由于 D_K与 D_1、.....、D_{K-1}相互独立：

$$I(\theta_K; D_1, \ldots, D_{K-1} \mid D_K) \;=\; I(\theta_K; D_1, \ldots, D_{K-1}) - I(D_K; D_1, \ldots, D_{K-1}; \theta_K) \tag{22}$$

其中，交互信息项存在上界。在最简单的情况下（任务完全独立且确定性U）：

$$I(\theta_K; D_K) + I(\theta_K; D_1, \ldots, D_{K-1}) \;\le\; I(\theta_K; D_1, \ldots, D_K) \;\le\; H(\theta_K) \;\le\; C. \tag{23}$$

保护参数 λ 控制着分配：较大的 λ 会约束 θ_K，使其保持在 θ_{K-1}附近，从而保留过去的信息，但会降低 $I(\theta_K; D_K)$。较小的 λ 允许自由适应 D_K，但会覆盖过去的信息。无论哪种极端情况，都无法超越总容量 C。

□

特殊情况：经验回放。重放通过维护一个存放过去样本示例的缓冲区 $\mathcal{B}$，部分打破了因果约束。这实际上将步骤 k时的"可访问信息"从 $I(\theta_{k-1};\ D_1、\ldots\ldots、\ D_{k-1})$（压缩到参数中的信息）增加到 $I(\theta_{k-1};\ D_1、\ldots\ldots、\ D_{k-1}) + I(\mathcal{B};\ D_1、\ldots\ldots、\ D_{k-1})$（参数加原始数据）。重放通过提供补充参数容量的外部记忆，实现了85.9%的准确率。不过，它仍有 $C_{\text{ctx}} = 0$：测试时所有任务都使用相同的 θ_K，因此P5协议的结果为 $\Delta_{\text{P5}} = 0$。

2.5.2 范式 B：状态变换（$C_{\text{ctx}} \to 0$）

在这一范式中，存在一个上下文信号 c_k，它与参数状态相结合，以生成任务适配参数：

$$\theta_k \;=\; f(\theta_{k-1}, c_k), \tag{24}$$

其中，f 是一种学习式变换（例如 CFlow 中的神经常微分方程）。与范式 C 的关键区别在于，f 将前一状态 θ_{k-1} 和上下文 c_k 都作为输入。原则上，这使上下文能够调节该变换。实际上，会出现以下结构性失效模式。

注 2（维度不匹配和上下文绕过）。当$\dim(\theta) \gg \dim(c)$ 且 f 被参数化为一个以拼接 $[\theta; c]$为输入的神经网络时，优化器通常会学会将任务信息编码在 θ（高维输入）中，而不是 c（低维输入）中。这是因为梯度幅值随输入维度变化：$\|\partial f/\partial\theta\| \propto \sqrt{d}$ 而 $\|\partial f/\partial c\| \propto \sqrt{m}$，其中 $m = \dim(c) \ll d = \dim(\theta)$。

我们的 CFlow 系统体现了这种失效模式。CFlow 使用神经常微分方程，其中流网络将拼接后的 $[\theta \in \mathbb{R}^{4842};\ c \in \mathbb{R}^{32}]$ 作为输入——维度比为 $\sim$150:1。尽管存在显式的上下文编码器，CFlow 仍表现出 $\Delta_{\text{P5}} = 0.0$，这证实了上下文信号在结构上对动力学不可见。CFlow 的 92.4% 准确率完全来自元学习初始化θ_0，而非上下文相关参数生成。用 C_{ctx} 的术语来说：

$$C_{\text{ctx}}^{\text{CFlow}} \;\approx\; 0 \quad (\text{structurally, despite } c \neq \emptyset). \tag{25}$$

这是一个具体例子，说明了拥有上下文信号与使用上下文信号之间的区别。C_{ctx} 框架正确地将 CFlow 判定为实际上不依赖上下文，尽管其架构表面上看似依赖上下文。

2.5.3 范式 C：条件再生（$C_{\text{ctx}} \gg H(T)$）

在这一范式中，预测参数由上下文条件生成器从头生成：

$$\theta_k \;=\; g(c_k), \tag{26}$$

其中，$g : \mathcal{C} \to \Theta$ 是一个学习式生成器，用于将上下文映射为参数。关键的结构属性是，θ_{k-1} 到 θ_k之间不存在通路——第 k 步的参数仅取决于上下文 c_k，而不取决于任何先前状态。这使 C_{ctx} 成为任务信息到达预测参数的唯一信道。

对于架构为 $\theta_k = \theta_{\text{base}} + V \cdot h(c_k)$ 的 HyperNetwork，其中 $V \in \mathbb{R}^{d\times r}$ 是低秩投影矩阵，$h:\ \mathcal{C} \to \mathbb{R}^r$ 是多层感知机，其上下文信道容量满足：

$$C_{\text{ctx}} \;\approx\; r_{\text{eff}}(V) \cdot b \quad \text{bits}, \tag{27}$$

其中，$r_{\text{eff}}(V) = \exp(-\sum_i \bar{s}_i \log\ \bar{s}_i)$ 是 V 的有效秩（根据其归一化奇异值$\bar{s}_i = \sigma_i / \sum_j \sigma_j$计算得出），$b$ 是每维有效比特。对于名义秩为 $r = 64$ 的 HyperNetwork，我们测得 $r_{\text{eff}} \approx 58.5$–59.1，得到 $C_{\text{ctx}} \approx 59 \cdot b \gg H(T) = \log_2(5) \approx 2.32$ 比特。

2.6 为什么超网络绕过不可能三角

HyperNetwork（von Oswald 等人，2020）似乎违反了定理 3：它们在在线评估中以有界参数实现了零遗忘。其原因在于，HyperNetwork改变了θ 的语义，从而绕开了该定理的假设。

状态与函数值。 在范式 A（随机梯度下降、EWC、突触智能）中， θ_k 是一个状态：它通过序贯更新进行维护，并且必须在测试时同时服务于所有任务。不可能三角之所以适用，是因为这一状态的容量 C受到上界限制。

在范式 C（HyperNet）中， $\theta_k = g(c_k)$ 是一个函数值：它会在每次评估时根据上下文重新计算，无需同时编码所有任务。“知识”并不存储在 θ_k 中，而是存储在生成器 g 的元学习参数 $\phi = (V,\ h,\ \theta_{\text{base}})$ 中。这些元参数 ϕ 通过情节式元学习在所有任务上进行联合训练，而不是按序训练。

联合与 。序贯优化。 元参数 ϕ通过目标函数进行优化

$$\phi^* \ = \ \arg\min_{\phi} \ \sum_{k=1}^{K} \mathcal{L}_k(g_\phi(c_k))\,, \tag{28}$$

其中， $\mathcal{L}_k$ 是任务 k 上的损失， g_ϕ 表示由 ϕ参数化的生成器。这是对所有任务进行的联合 优化——因果约束不适用，因为 ϕ 在训练期间可以看到所有任务。不可能三角关于序贯更新（$\theta_k = U(\theta_{k-1},\ D_k)$）的假设在结构上并不适用。

形式化分析。让我们比较信息流：

$$\text{Paradigm A:} \quad D_1 \ \to \ \theta_1 \ \to \ \cdots \ \to \ \theta_K \quad \text{(Markov chain, DPI applies)} \tag{29}$$

$$\text{Paradigm C:} \quad \{D_1, \ldots, D_K\} \ \to \ \phi^* \ \to \ \theta_k = g_{\phi^*}(c_k) \quad \text{(no Markov chain)} \tag{30}$$

在范式 A 中，数据处理不等式保证信息损失是单调的： $I(\theta_K;\ D_1) \ \le \ I(\theta_1; D_1)$。在范式 C 中，不存在这样的链条——每个 θ_k 都由相同的元参数 ϕ^*独立地生成，而这些元参数同时编码了所有任务的信息。信息 $I(\theta_k;\ D_k)$不受任何序贯瓶颈的约束；它仅受 C_{ctx} （有多少任务特定信息流经上下文）以及 g的表达能力的约束。

“不可绕过性”原则。上述分析揭示了为何架构设计比算法复杂度更为重要。在范式 C 中，上下文通路是任务信息到达 θ_k的唯一路径。不存在可作为替代信息通道的高维状态 θ_{k-1} 。这种“不可绕过性”恰恰迫使模型使用上下文——从而获得较高的 C_{ctx}。

将此与 CFlow（范式 B）进行对比：在其中，上下文 c 与 4842 维状态 θ拼接在一起。在这里，状态提供了宽得多的替代信道，因此优化器会理性地忽略狭窄的上下文通路。结果是： $\Delta_{\text{P5}} = 0.0$ （未使用上下文），尽管架构名义上包含上下文输入。

注 3 （与注意力机制和 RNN 类比的联系）。 范式 *A* 与范式 *C* 的关系，映照了循环神经网络与注意力机制在序列建模中的关系。*RNN* 会维护一个隐藏状态 $h_t = f\ (h_{t-1}, x_t)$ ，但会受到梯度消失的影响——这是灾难性遗忘在序列层面的对应现象。注意力机制通过基于内容的寻址直接访问过去的输入，从而绕过序贯瓶颈。同样，超网络通过基于上下文的寻址直接生成任务特定参数，从而绕过序贯参数瓶颈。在这两种情况下，解决方案都是以直接条件生成取代序贯状态压缩。

3 通过 C_{ctx} 视角对失效进行分类

C_{ctx} 框架为分析持续学习方法为何会失败提供了统一的术语体系。在本节中，我们系统地考察四种失败范式——状态保护、受生物学启发的学习、

列专门化和状态变换——并说明它们都可归结为违反上下文信道容量界（定理4）。本分析基于为期86天、涵盖 15+ 个不同研究方向的1,130+项实验。

3.1 范式A：状态保护方法

状态保护方法试图通过约束新任务到来时 θ 的变化方式来防止遗忘。它们具有共同的结构属性：不存在上下文通路，因此根据定义有 $C_{\text{ctx}} = 0$ 。因此，定理4预测，无论具体的约束机制如何，遗忘都将达到最大。

3.1.1 基于正则化的方法（EWC、SI、LwF）

机制。 弹性权重巩固（Kirkpatrick等人，2017）增加惩罚项 $\frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_i^*)^2$，其中$F_i$ 是在先前任务上计算得到的参数 i 的对角费舍尔信息。突触智能（Zenke等人，2017）在训练过程中累积在线重要性度量 $\Omega_i \propto \sum_t g_i(t)\,\Delta\theta_i(t)$。无遗忘学习（Li和Hoiem，2017）通过$\mathcal{L}_{\text{KD}} = \alpha\,\text{KL}(\sigma(\hat{y}_{\text{old}}/T) \,\|\, \sigma(\hat{y}_{\text{new}}/T))$将旧模型的软输出蒸馏到新模型中。

C_{ctx} **分析**。 这三种方法都使用单一共享参数向量 θ，且没有任何机制能在推理时根据任务身份进行条件化。更新规则 $U(\theta_{k-1},\ D_k)$ 会原地修改 θ，测试时所有任务使用的都是同一个 θ_K 。因此 $C_{\text{ctx}} = 0$，定理4预测遗忘不可避免。

实验结果。 在 Split-MNIST 上（表1）：

- EWC： 18.9 ± 0.1% 准确率，遗忘率 97.6% $\approx$ NaiveSGD： 18.7 ± 0.3% 准确率，遗忘率 97.1%。
- 突触智能： 16.4 ± 3.7% 准确率，遗忘率 97.1%——比 NaiveSGD更差 。
- LwF： 24.2 ± 0.6% 准确率，遗忘率 54.8%——知识蒸馏提供了一定保护，但无法克服 $C_{\text{上下文}} = 0$。

根本原因。对于 EWC 和突触智能，失败并不仅仅是因为 $C_{\text{ctx}} = 0$；正则化本身也不起作用。多层感知机架构（784→256→128→10，约 235K 个参数）的参数空间过大，以至于对角Fisher近似无法对其施加有意义的约束。Fisher信息矩阵为 235K $\times$ 235K；对角近似舍弃了所有非对角结构，使有效约束退化为逐参数惩罚，而优化器只需沿损失景观中的低费舍尔方向移动，就能轻易绕过这些惩罚。结果： $|\text{准确率}_{\text{EWC}} - \text{准确率}_{\text{SGD}}| = 0.2$个百分点，处于噪声范围内。

LwF 的部分成功（遗忘率从 97% 降至 55%）可以用蒸馏损失提供了隐式“软上下文”来解释：旧模型的输出分布携带与任务相关的信息。然而，由于旧模型从未更新，这些信息会呈指数衰减，而且共享的 θ 在推理时仍有 $C_{\text{ctx}} = 0$。

3.1.2 Deep NeuralDrift(DND)—赫布零结果

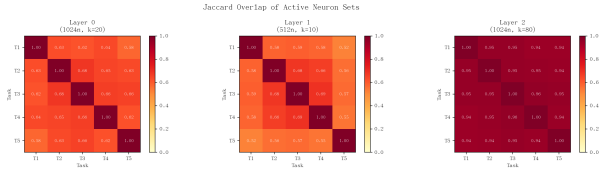
机制。 深度神经漂移（DND）是一种受生物学启发的架构，结合了Oja规则赫布学习、top-k 稀疏激活、共振门控和余弦相似度模板匹配。设计假设是，赫布学习能够发现任务判别特征，这些特征与稀疏激活相结合，将自然地在各任务之间划分表示空间。

研究规模。 我们在12天内进行了 $108+$ 项实验，系统探索了髓鞘化阈值、共振功率、稀疏范围、模板动态和列大小。最佳配置（S20：1024个隐藏单元，top-$k = 80$，并采用任务边界模板刷新）达到了73.19%的准确率，遗忘为 -0.42%（8个随机种子）。

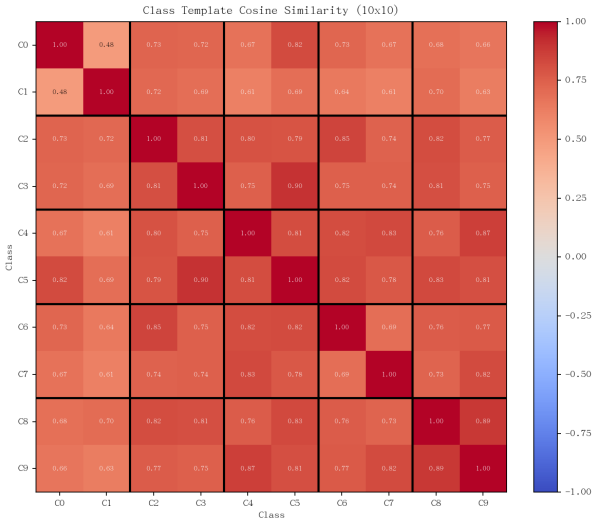
冻结随机对照。 关键的诊断性实验是将所有赫布权重冻结在随机初始化状态（可塑性率 $= 0$，输出学习率 $= 0$），同时保持其他所有条件完全相同：

$$\text{Hebbian-trained DND + TBTR + Final Pass:}\quad 80.85\%\ \text{ACC} \tag{31}$$

$$\text{Frozen random DND + TBTR + Final Pass:}\quad 81.95\%\ \text{ACC} \tag{32}$$



(a) 任务间神经元激活重叠（Jaccard指数）。输出层重叠 $= 0.947$；隐藏层 > 0.60。几乎所有神经元都被共享。



(b) 模板余弦相似度：任务间（0.751）$>$ 任务内（0.737）。任务间模板更相似，而非任务内更相似——这与专门化的预期相反。

图1：DND神经元重叠与模板相似度分析。未观察到涌现式任务专门化：神经元在任务间共享，模板也未能区分不同任务。

冻结随机网络的表现优于赫布训练网络 1.1 个百分点。赫布学习对持续学习表现的贡献恰好为零。

理论解释。这一结果源于赫布规则的一项众所周知的性质：

命题8（赫布学习收敛至PCA，改编自Oja 1982）。任何形式为$\Delta W \propto f(W, x, y)$的赫布学习规则，只要满足局部性和稳定性，就会收敛到由输入协方差矩阵 $\mathbb{E}[xx^\top]$的主特征向量张成的子空间——即主成分分析方向。

主成分分析捕捉的是全局 方差最大的方向，而非任务判别方向。在隐藏单元 $N = 1024$ 采用 top-$k = 80$ 激活的过参数化情形下，组合容量为：

$$C_{\text{comb}} = \log_2 \binom{1024}{80} \approx 290 \text{ bits} \gg H(Y) \approx 3.3 \text{ bits} \tag{33}$$

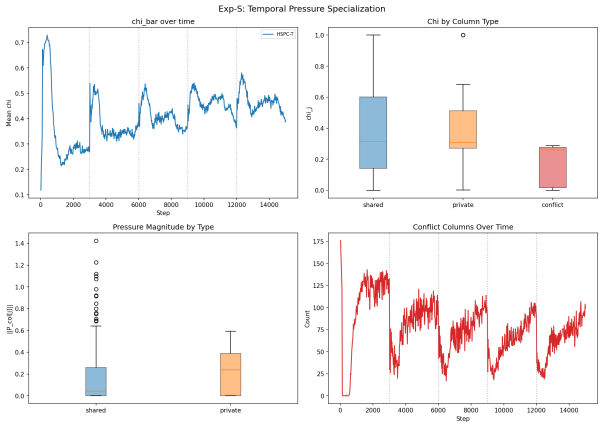
当 $C_{\text{comb}} \gg H(Y)$ 时，投影方向的具体选择并不重要——任何 足够不相干的方向集合，都能为分类任务提供足够的组合容量。随机特征与学习特征之间的准确率差异上界为：

$$|\text{ACC}_{\text{random}} - \text{ACC}_{\text{learned}}| \leq O\left(\frac{H(Y)}{C_{\text{comb}}}\right) \approx \frac{3.3}{290} \approx 0.011 \tag{34}$$

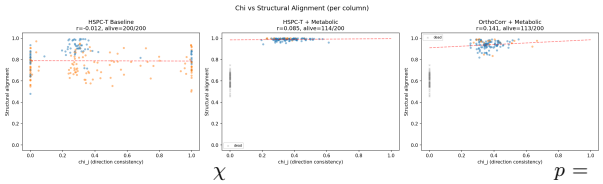
实验 观察到的差异 $|80.85 - 81.95| = 1.1$pp ≈ 0.011 与此上界一致 .

C_{ctx} **分析**。 DND 没有显式上下文通路。虽然稀疏激活模式会随输入而变化，但这种变化不可控——无论任务身份如何，相同的输入总会产生相同的激活模式。模板匹配分类器使用一组共享模板，不具备任务条件路由。因此 $C_{\text{ctx}} = 0$。

图 1 提供了直接证据：输出层的任务间 Jaccard 重叠度为 0.947（几乎所有神经元都被共享），且模板的任务间相似度（0.751）高于任务内相似度（0.737）——这与专门化所要求的情况恰恰相反。



(a) HSPC-T 实验中的列专门化指标。所有配置的任务特定列使用率都接近于零。



(b) 代谢 0.87 χ 与列对齐： $\rho = 0.018$。 $p =$ χ 统计量不包含任何结构信息。

图 2：支持 S_N 对称性障碍的证据。(a) HSPC-T 未能实现列专门化。(b) 代谢剪枝的 χ 统计量与列重要性不相关。

3.1.3 HSPC-T 列专门化—— S_N 不可能性

机制。带时间压力的分层稀疏预测编码（HSPC-T）假说认为，时间压力（紧迫性信号、代谢成本）可以驱动字典学习网络中的各个列专门化以处理不同任务，并通过列路由形成隐式上下文。

研究规模。我们在三个子方向开展了 40+ 项实验：时间压力调度、代谢剪枝（基于 χ 的权重衰减），以及 InfoMax 快速／慢速 EMA 选择性信号传递。

这种 S_N 对称性屏障。三个子方向都因同一个根本原因而失败：

命题 9 (S_N 重建损失的置换对称性)。 对于具有列 $\{(w_i, e_i)\}_{i=1}^N$、以重建损失 $\mathcal{L} = \|x - W_d z\|^2$ 为最小化目标的字典学习网络，其中 $z = \text{encode}(x; W_e)$，损失在对称群 S_N 作用下保持不变，该作用通过置换列索引实现：

$$\mathcal{L}(x; \{(w_{\sigma(i)}, e_{\sigma(i)})\}) = \mathcal{L}(x; \{(w_i, e_i)\}) \quad \forall \sigma \in S_N \tag{35}$$

因此，所有列 i的梯度场 $\nabla_{w_i}\mathcal{L}$ 在结构上完全相同，任何由损失驱动的机制都无法打破这种对称性，进而诱导任务专门化。

任何未显式注入列特定偏置（例如通过任务身份）的逐列扰动 $\mathcal{L}_{\text{col}}(i)$，其梯度贡献都会被主要的重建梯度 $O(\|x\|^2/N)$所主导。实证结果证实了这一点：

- **代谢剪枝**：用于识别与任务相关列的 χ 统计量与真实列重要性之间的斯皮尔曼$\rho = 0.018$ ($p = 0.87$)相关性为零，完全不具备判别能力。激活频率（$\rho = 0.916$）才是真正的区分指标，但自适应阈值已经能够处理这一问题。

- **M1 InfoMax**：采用快速/缓慢指数移动平均信号和 $\eta \in \{0.001, 0.01, 0.1\}$ 学习率，均未能实现列专门化（在任何 η验证指标中，0/4 项通过）。

C_{ctx} 解释。列专门化旨在通过网络内部动力学创造涌现式的上下文。 S_N 对称性结果表明，若无显式的对称性破缺，这一点便无法实现——而显式上下文通路恰恰提供了这种破缺。

3.2 ParadigmB：状态变换——CFlow ODE

机制。CFlo w（NeuralDrift团队，2026）将参数轨迹建模为连续时间常微分方程 :

$$\frac{d\theta}{dt} = f_\phi(\theta, c, t) \tag{36}$$

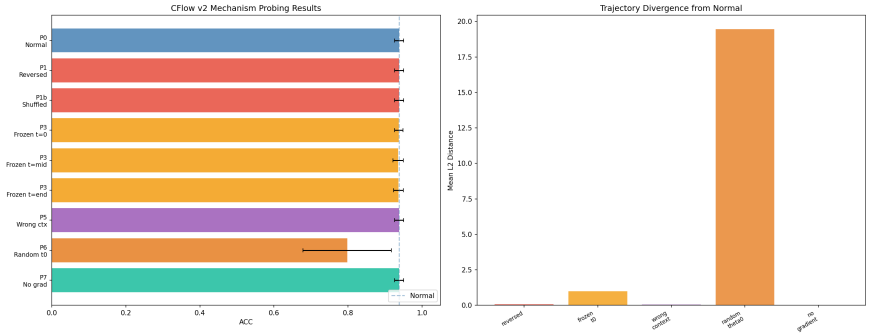


图 3：CFlow 探测结果。P5（错误上下文）不会导致准确率发生任何变化，而 P6（随机 θ_0）则会使准确率下降 -40 个百分点。这证实 CFlow 是一个“θ_0 记忆器”——上下文通路在结构上已失效。

其中，$c \in \mathbb{R}^{32}$ 是学习式上下文向量，而 f_ϕ 由神经网络（“流网络”）参数化。给定任务 k的上下文 c_k，系统从共享初始化 θ_0 开始积分，以生成任务特定参数 $\theta_k = \theta_0 + \int_0^1 f_\phi(\theta(t), c_k, t)\, dt$。

表面结果。 CFlow 在 Split-MNIST 上达到 $92.4 \pm 0.7\%$ 的准确率，遗忘率仅为 6.1%，远优于 EWC/SI/LwF，且与经验回放不相上下。这似乎验证了常微分方程方法。

P5 探测揭示了真相。 然而，错误上下文探测带来了一个令人震惊的结果：

$$\Delta_{\text{P5}}^{\text{CFlow}} = 0.0\text{pp} \tag{37}$$

无论模型接收哪个上下文向量，其准确率都完全相同。上下文编码器被彻底忽略了。

根本原因：元学习旁路塌缩。 流网络接收的输入为 $[\theta; c] \in \mathbb{R}^{4874}$ 其中 $\dim(\theta) = 4842$ 且 $\dim(c) = 32$，形成了 $\approx 150{:}1$ 的维度比。我们发现，有两种机制共同作用，导致上下文不可见：

命题10 (梯度幅度不对称)。 对于一个接收拼接输入 $[\theta; c]$ 的神经网络 f，其中 $\dim(\theta) = d_\theta \gg \dim(c) = d_c$，其期望梯度幅值满足：

$$\frac{\mathbb{E}\left[\|\nabla_{\theta_0}\mathcal{L}\|\right]}{\mathbb{E}\left[\|\nabla_c\mathcal{L}\|\right]} = O\left(\sqrt{\frac{d_\theta}{d_c}}\right) \approx \sqrt{\frac{4842}{32}} \approx 12.3 \tag{38}$$

当参数以相同尺度初始化时。优化器遵循 θ_0 梯度，使 θ_0 成为任务编码状态，而非任务中立的初始化。

P6验证。 将 θ_0 替换为随机初始化（P6探测）会使准确率下降40个百分点，证实所有任务信息都存在于元学习得到的 θ_0中：

$$\Delta_{\text{P6}}^{\text{CFlow}} = -40\text{pp} \quad \Longrightarrow \quad \text{performance is entirely } \theta_0\text{-dependent} \tag{39}$$

C_{ctx} 分析。 CFlow原则上具有上下文通路（理论上）（c 是 f_ϕ的输入），但拼接架构使其可被绕过。优化器利用 θ 通路编码任务信息，降低$C_{\text{ctx}}^{\text{effective}} \approx 0$，尽管有 $C^{\text{architectural}}{}_{\text{上下文}} > 0$。这就是“上下文隐式”失效模式：上下文虽存在，但在结构上不可见。

3.3 其他闭合方向

表12总结了本项目期间探索的 $15+$ 其他研究方向，但这些方向均未成功。C_{ctx} 框架对每个方向都作出了解释。

表 3：已关闭研究方向总结。所有失败最终都归结为 $C_{\text{ctx}} = 0$ 或上下文利用的结构性障碍。这些方向共进行了 200+ 次实验。

方向	实验次数	ACC	根本原因
空间能量	18	±0pp	距离加权的 L_1 = 正则化；在容量压力下效果不佳
河床效应	12	< 0个百分点	均匀抑制 $-\Sigma a_i \log \pi_i$，而非选择性路由
SPC-TC（ISTA）	82	−0.6个百分点	信息瓶颈；学习字典 ≈ 随机字典
CCD理论	20	N/A	定性上颇具启发性，但定量上不成立（5项预测中有3项错误）
OrthoCorr	8	+0.3%	与自适应阈值相冲突；当 $k_{\text{eff}} \geq D$ 时无法提供帮助
代谢剪枝	16	3/5 失败	χ 没有判别能力（$\rho = 0.018$）
M1 InfoMax	12	0/4 通过	无法打破 S_N 对称性

3.4 “冻结式 >学习式”模式

在我们的实验中反复出现的一种模式是：在持续学习场景下，冻结随机特征的表现与学习特征相当，甚至更优：

1. **DND**：冻结随机 81.95% > 赫布训练 80.85%（方程式 31–32）。

2. **SPC-TC** ：冻结随机字典 ≈ 在线学习字典（对齐差异 $= 0.03$).

3 。**CIFAR-10 上下文**：随机像素特征 45.2% >学习得到的CNN特征 37.6 %.

C_{ctx} 框架提供了统一的解释。在组合容量 $C_{\text{comb}} \gg H(Y)$ 的情况下，Johnson-Lindenstrauss引理保证随机投影以高概率保留成对距离。学习并不能提升特征质量，却会引入稳定性惩罚：对共享特征进行梯度更新会导致不同任务之间产生干扰。冻结特征则完全避免了这种惩罚。信息论界为：

$$\text{Stability gain from freezing} \geq \text{Quality gain from learning} \quad \text{when } C_{\text{comb}} \gg H(Y) \tag{40}$$

这有一个惊人的含义：在过参数化持续学习系统中，特征提取器应保持冻结，所有适应都应通过上下文条件通路进行。这正是我们的超网络架构（第 6 节）的设计。

4 错误上下文探测协议

我们将第 2.4 节介绍的错误上下文探测（P5）协议形式化，并将其扩展为一系列诊断探针，以共同刻画持续学习架构如何使用（或未能使用）上下文信息。

4.1 协议定义

定义 6 （P5 探测系列）。 对于上下文条件化 *CL* 模型，其上下文编码器为 e，参数生成器为 g，基础参数为 θ_{base}，我们定义以下探针：

P5a：错误的任务身份 （针对预言机上下文）。使用上下文 $c_{(k+1) \bmod K}$ 评估任务 k 。测量：预言机任务 *ID* 是否确实被使用。

P5b：随机上下文向量。使用 $c \sim \mathcal{N}(0,\ I)$ 评估任务 k。测量：任何 上下文信息（而不仅仅是正确的上下文信息）是否会产生影响。

P6：随机基础参数。替换为 θ_{base} 随机初始化，保留正确上下文。衡量：对元学习初始化的依赖。

P 7：零上下文。使用 $c = \mathbf{0}$进行评估。衡量： θ_{base} 单独的基线性能。

对于每个探测 P_i，我们报告 $\Delta_{P_i} =$ 准确率$_{P_i}$ −准确率$_{\text{normal}}$。

解释 **网格**。探测结果的组合揭示了架构的信息路由 :

- $\Delta_{\mathrm{P5}} \approx 0, \Delta_{\mathrm{P6}} \ll 0$: “$\theta_0$ 记忆器”（CFlow 模式）。上下文已失效；所有信息都在初始化中。
- $\Delta_{\mathrm{P5}} \ll 0, \Delta_{\mathrm{P6}} \approx 0$: “上下文相关”（HyperNet 模式）。性能完全来自上下文条件参数生成。
- $\Delta_{\mathrm{P5}} \approx 0, \Delta_{\mathrm{P6}} \approx 0$: 上下文与初始化均无关（退化情形）。

4.2 C_{ctx} 代理估计

我们将实证上下文信道容量代理指标定义为:

$$\hat{C}_{\mathrm{ctx}} = \max(0,\ -\Delta_{\mathrm{P5}}) \tag{41}$$

该代理指标的取值范围为 $[0, 1]$: 值为 0 表示完全忽略上下文（$C_{\mathrm{ctx}} = 0$），而接近 1 的值则表示完全的上下文依赖（$C_{\mathrm{ctx}} \gg H(T)$）。

该代理指标的局限性。基于 P5 的代理指标提供的是二元 信号（是否使用上下文），而非以比特为单位对 C_{ctx} 进行连续估计。$\hat{}$ C_{ctx} 的中间值（例如 0.3–0.7）难以解释。我们将在第7节讨论使用互信息神经估计进行改进的可能性。

4.3 有效秩作为容量度量

对于具有显式参数生成矩阵的架构（例如 HyperNet 的 $V \in \mathbb{R}^{d_\theta \times r}$ 将上下文映射为参数增量），我们通过奇异值分解定义一种额外的容量度量。

定义 7（有效秩）。给定奇异值为 $\sigma_1 \geq \cdots \geq \sigma_r$ 的奇异值分解 $V = U\Sigma W^\top$，定义归一化奇异值 $\bar{\sigma}_i = \sigma_i / \sum_j \sigma_j$。有效秩为:

$$r_{\mathrm{eff}} = \exp\left(-\sum_{i=1}^{r} \bar{\sigma}_i \log \bar{\sigma}_i\right) = \exp(H(\bar{\sigma})) \tag{42}$$

基于秩的容量估计为 $C_{\mathrm{ctx}}^{\mathrm{rank}} = r_{\mathrm{eff}} \times b$ 比特，其中 b 是每维有效比特。

对于我们的 HyperNet，跨随机种子为 $r_{\mathrm{eff}} \approx 58.5$–59.8（名义秩为 $r = 64$），表明上下文通道的利用率接近满载。即使保守地估计每维 $b = 1$ 比特，$C_{\mathrm{ctx}}^{\mathrm{rank}} \approx 59$ 比特$\gg H(T) = \log_2 5 \approx 2.3$ 比特。

5 实验设置

5.1 基准测试

Split-MNIST（主要基准）。MNIST 数字被划分为 5 个依次进行的二分类任务: $\{0, 1\}$、$\{2, 3\}$、$\{4, 5\}$、$\{6, 7\}$、$\{8, 9\}$。所有方法共用一个包含 10 个类别的输出头，并在最后一个任务训练完成后，对全部 10 个类别进行评估。每个类别使用 500 个训练样本和 200 个测试样本。

Split-CIFAR-10（扩展基准）。CIFAR-10 图像（$32 \times 32 \times 3$）被划分为 5 个任务，每个任务包含 2 个类别。该基准的难度显著更高，因为不同任务的批次级像素统计几乎相同（余弦相似度 > 0.995），这给学习式上下文编码器带来了严峻挑战。

5.2 架构细节

表 4 给出了所有方法的完整架构规格。

无上下文基线使用一个包含 256 和 128 个隐藏单元的两层多层感知机，采用 ReLU 激活函数，并输出 10 类 softmax。所有模型均使用 Adam优化器，学习率为 10^{-3}。

表4：所有评估方法的架构与超参数详情。

类别	方法	架构	关键超参数
Ctx-free	NaiveSGD	多层感知机(784→256→128→10)	学习率 $=10^{-3}$，Adam
	EWC	相同的多层感知机	$\lambda = 400$
	SI	相同的多层感知机	$c = 1.0$
	LwF	相同的多层感知机	$\alpha = 1.0$, $T = 2.0$
	经验回放	相同的多层感知机	缓冲区 $=200$/任务
隐式上下文	CFlow	ConvTaskNet（4 个滤波器） + FlowNet	$\theta_{\text{dim}} = 4842$, $c_{\text{dim}} = 32$
Ctx	HyperNet 预言机	ConvTaskNet(4f) + HyperNet	上下文$=64$d，秩 $=64$
	HyperNet 学习式	相同 +学习型编码器	直接_统计数据，cw $=0.1$，bd $=0.3$

CFlow 使用一个小型 CNN骨干网络（ConvTaskNet，包含 4 个卷积滤波器，可生成一个 4842 维的参数向量 θ）和一个包含 256 个隐藏单元的三层流网络。上下文编码器生成 32 维的上下文向量。元学习使用 80 个回合，重放比例为 0.3，每个任务使用 100 个重放样本。

超网络 使用相同的 CNN骨干网络。HyperNet 通过 $\theta_k = \theta_{\text{base}} + V \cdot g(c_k)$生成参数增量，其中 $V \in \mathbb{R}^{4842\times 64}$ 是低秩投影矩阵，g：$\mathbb{R}^{64} \to \mathbb{R}^{64}$ 是一个含 256 个隐藏单元的两层多层感知机。**HyperNet Oracle** 接收独热任务身份作为上下文。**HyperNet Learned**使用批次统计编码器，并启用 direct_stats=True（当前批次均值/方差，完全可微），通过对比损失（权重 0.1）区分任务上下文，并使用基础参数丢弃（比率 0.3）防止θ_{base} 绕过。联合多任务训练使用 200 个回合。

5.3 训练协议

- **无上下文基线**：每个任务训练 5 个轮次，按顺序依次在任务 $1 \to 5$上训练。经验回放从缓冲区中均匀抽取样本；该缓冲区为每个过去任务存储了 200 个样本。
- **CFlow**：进行 80 个元学习回合。每个回合随机抽取一种任务顺序，训练 2 个轮次，并评估 3 个轮次。流网络以元学习率 10^{-3}进行端到端训练。
- **超网络**：200 个联合多任务回合。每个回合采样一个随机任务，从上下文生成任务特定参数，并更新 HyperNet 参数（但不 θ 直接）。这种联合训练在构造上避免了序列遗忘。

所有实验均使用 4 个随机种子（$\{0, 1, 2, 3\}$），并报告平均值 $\pm$ 标准差。所有代码、配置和结果均可在补充材料中获取。

5.4 评估指标

- **准确率**：在最终任务上完成训练后，全部 5 个任务的平均准确率：准确率 $= \frac{1}{K}\sum_{k=1}^{K} a_k^K$。
- **遗忘（遗忘）**：平均最大准确率下降：遗忘 $= \frac{1}{K-1}\sum_{j=1}^{K-1}$（最大$_{t\le K}\, a_j^t - a_j^K$）。
- **后向迁移（后向迁移）**：后向迁移 $= \frac{1}{K-1}\sum_{j=1}^{K-1}$（$a_j^K - a_j^j$）。负的后向迁移表示遗忘；正的后向迁移表示旧任务上有所提升。
- **P5** Δ：错误上下文探测差值（定义 6）。
- $\hat{C}_{\text{上下文}}$：上下文信道容量代理指标 $=\max(0,\ -\Delta_{\text{P5}})$。
- C_{rankctx}：HyperNet 的 V 矩阵的有效秩（定义 7）。

6 结果

6.1 主要结果

表1（为方便起见在此重现）展示了完整的 Split-MNIST 结果。主要观察结果如下：

观察 1：EWC ≈ 突触智能 ≈ NaiveSGD。 Fisher加权正则化（EWC）和路径积分重要性（SI）几乎无法提供保护——它们的准确率（18.9% 和 16.4%）与朴素微调（18.7%）在统计上无法区分。对于一个含 235K 个参数的多层感知机而言，对角Fisher近似过于粗糙，无法对优化施加有意义的约束。

观察 2：LwF 提供部分保护。 知识蒸馏将遗忘率从 97% 降至 55%，使准确率提升至 24.2%。这与软蒸馏目标每个样本携带约 $\log_2(10) \approx 3.3$ 比特的任务相关信息这一点相符，但随着教师模型被覆盖，这些信息会逐渐衰减。

观察 3：重放效果显著。 经验回放仅使用每个任务存储的 200 个样本，就达到了 85.9% 的准确率，遗忘率仅为 12.5%。这表明，即使缓冲区很小，打破因果约束（重新访问旧数据）也非常有效。从信息论的角度来看，在每个训练步骤中，重放都会注入 $I(D_{\text{buffer}}; D_{\text{old}})$ 比特的历史任务信息。

观察 4：CFlow 的误导性表现。 CFlow 达到 92.4% 准确率——优于重放——但其 $\Delta_{\text{P5}} = 0.0$ 表明，这一表现完全来自元学习初始化θ_0，而非上下文条件常微分方程。CFlow 实际上是一种精密的元学习算法，而不是上下文条件持续学习方法。

观察 5：HyperNet 实现完美的持续学习。 两种 HyperNet 变体均达到 $\geq 98.8\%$ 准确率，遗忘为 0.0%。-97.6 百分点（预言机）和 -95.2 百分点（学习式）的 P5 变化量证实，性能完全依赖于上下文。这是唯一一个 $C_{\text{ctx}} > 0$的架构族。

6.2 C_{ctx} 对比遗忘：二元相变

ˆ 图 ?? 针对全部 8 种方法绘制了 C_{ctx} 与遗忘的关系。这种关系并非渐进式的——它呈现出二**元相变**：

- **阶段 I (** $\hat{C}_{\text{ctx}} = 0$**)**：所有无上下文方法与 CFlow 均聚集在高遗忘区间（6–97%）。在此阶段内，遗忘程度因具体学习算法而异（重放 $<$ 蒸馏 $<$ 正则化$<$ 朴素），但所有方法都共享 $C_{\text{ctx}} = 0$ 这一根本局限。

- **第二阶段 (** $\hat{C}_{\text{上下文}} \approx 1$**)**：两个 HyperNet 变体都实现了零遗忘。转变十分陡峭——ˆ不存在遗忘程度居中的方法，其 $C_{\text{上下文}}$ 取值也没有处于中间范围。

这种二元模式由定理4预测：遗忘的下界为$(1 - C_{\text{ctx}}/H(T)) \cdot$ 遗忘$_{\max}$，它会从 Fgt$_{\max}$ （当 $C_{\text{ctx}} = 0$时）陡然转变为 0（当 $C_{\text{ctx}} \geq H(T)$时）。中间区间没有方法，并非抽样造成的假象，而是反映了实现 $0 < C_{\text{ctx}} < H(T)$的结构性困难。上下文通路要么在架构上无法绕过（得到 $C_{\text{ctx}} \gg H(T)$），要么优化器绕过该通路（得到 $C_{\text{ctx}} \approx 0$）。

6.3 错误上下文探测分析

图?? 以分组柱状图展示了 P5 探测结果（正常上下文与错误上下文准确率）。上下文条件方法的完整探测表如下：

CFlow 的上下文盲视。 在全部 4 个随机种子中，CFlow 的 P5 准确率与常规准确率在数值精度范围内相等。结合 P6 准确率 $= 52.4\%$（较随机 θ_0 低 -40 个百分点），这证实了“θ_0记忆器”模式：所有任务判别信息都存在于元学习初始化中，而流网络会忽略上下文编码器的输出。

HyperNet 的上下文依赖性。 对于 HyperNet Oracle，错误上下文会使准确率从 98.8% 降至 1.2%——对于 10 个类别而言接近随机猜测水平。随机上下文（P5b）得到 10.4%（接近随机猜测水平），这证实不存在某个恰好有效的单一随机上下文。P6 得到 49.7%，表明 θ_{base} 单独即可提供一个合理但远非最优的起点。

表5：上下文条件方法的完整探测结果（4个随机种子）。P5 = 错误上下文，P5b =随机上下文，P6 = 随机 θ_{base}，P7 = 零上下文。

方法	常规准确率	P5 准确率	P5b 准确率	P6 准确率	P7 准确率	$\hat{C}_{\text{ctx}}$
CFlow	92.4	92.4	—	52.4	—	0.000
HyperNet 预言机	98.8	1.2	10.4	49.7	—	0.976
HyperNet 学习式	98.9	3.7	19.5	—	53.1	0.952

表6：CIFAR-10 扩展结果（4个随机种子）。批次统计量失效；梯度上下文恢复性能；NestedCapsule 达到新的 SOTA。

方法	准确率（%）	遗忘率（%）	P5 Δ	上下文来源
HyperNet Oracle	77.7 ± 0.9	0.0	-77.0	独热任务标识
学习型超网络（批次）	54.4 ± 2.1	—	-0.5	批次像素均值/方差
HyperNet + 梯度上下文	77.0 ± 0.9	0.0	-77.0	$\nabla_\theta \mathcal{L}$
NestedCapsule +梯度上下文	$\mathbf{78.5 \pm 0.6}$	0.0	-78.5	$\nabla_\theta \mathcal{L}$ + 路由
NestedCapsule Oracle	79.5 ± 0.9	0.0	—	独热任务标识

ˆHyperNet Learned 的编码器质量。 学习到的编码器达到 $C_{\text{ctx}} = 0.952$——略低于 Oracle 的 0.976，但对于实现零遗忘而言仍然足够。P5b（随机上下文）的结果为 19.5%，高于 Oracle 的 10.4%，这表明学习到的编码器的上下文空间分布可能不如独热编码均匀。零上下文（P7）的准确率为 53.1%，说明 θ_{base} 承载了有意义但不完整的任务信息。

6.4 准确率矩阵分析

图?? 展示了完整的 5×5 准确率矩阵。每一行 i 表示在任务i上训练后的评估结果；每一列 j 表示在任务 j上的准确率。这些矩阵呈现出四种明显不同的视觉模式：

1. **对角线渐暗**（NaiveSGD、EWC、突触智能）：只有对角线元素明亮——模型仅保留最近训练的任务。先前任务的表现会立即衰减至接近零。2. **部分保留**（LwF）：对角线明亮，先前任务则有部分保留，但会随时间逐渐衰减。任务1的准确率在随后4个任务中从100%降至 $\sim 20\%$。3. **中度保留**（重放）：大多数元素亮度中等（$> 70\%$），早期任务的表现有所下降。回放缓冲区可防止完全遗忘。4. **整体高亮**（HyperNet）：所有元素亮度均匀且较高（$> 97\%$）。每一行都完全相同，因为HyperNet会根据上下文即时生成任务参数，不受训练历史影响。

CFlow 矩阵值得注意的是，尽管它采用了上下文编码器，其表现模式却更接近重放（保持能力中等），而非 HyperNet。这一视觉证据支持 P5 的发现：CFlow 的性能源自元学习初始化，而非上下文条件生成。

6.5 扩展至 CIFAR-10

我们将分析扩展到 Split-CIFAR-10，以检验该框架在 MNIST 之外的适用性。这个基准揭示了批次统计上下文编码器的一项关键局限，并促使我们提出一种新颖的解决方案。

上下文坍塌问题。 在 CIFAR-10 上，曾在 ˆMNIST ($C_{\text{ctx}} = 0.952$) 上表现完美的批次统计编码器却遭遇了灾难性失效。原因可以量化：CIFAR-10 各任务划分之间批次像素统计量（平均值、方差）的余弦相似度超过 0.995。对于五个任务，上下文向量几乎完全相同，导致 $C_{\text{ctx}}^{\text{effective}} \approx 0$ 降低。

梯度上下文编码器。 为解决上下文坍塌问题，我们提出使用相对于 θ_{base} 的损失梯度 作为上下文信号：

$$c_k = \text{MLP}(\Pi_{\text{rand}} \cdot \nabla_{\theta_{\text{base}}} \mathcal{L}(x_k, y_k; \theta_{\text{base}})) \tag{43}$$

其中 $\Pi_{\text{rand}} \in \mathbb{R}^{128 \times d_\theta}$ 是一个用于降低梯度维度的冻结的随机投影矩阵，多层感知机则映射到64维上下文空间。

关键洞见在于，使用真实标签 计算的梯度会在各任务之间产生近乎正交的上下文向量：

$$\cos(\nabla_\theta \mathcal{L}_{\text{task } i},\ \nabla_\theta \mathcal{L}_{\text{task } j}) \approx -0.19 \quad (i \neq j) \tag{44}$$

而伪标签梯度（使用模型自身的预测）则产生几乎相同的向量（cos ≈0.95），这解释了自监督上下文失败的原因。

结果。 梯度上下文编码器在Split-CIFAR-10上以 $\Delta_{\text{P5}} = -77.0$百分点达到 77.0 ± 0.9%准确率——将预言机差距从23.3百分点（批次统计量）缩小到仅0.7百分点。各随机种子的准确率（77.5%、78.1%、76.4%、75.9%）高度稳定。

嵌套胶囊超网络。 我们通过用 $N = 8$ 个胶囊记忆模块替换HyperNet的单个多层感知机来进一步提升性能，每个模块产生一个秩为8的部分向量。胶囊路由对相似度、幅值以及受Titans启发的惊奇信号使用Softmax注意力：

$$\alpha_i = \text{softmax}\left(\frac{s_i \cdot m_i \cdot (1 + \text{surprise}_i)}{\tau} \right) \tag{45}$$

其中， s_i 是上下文与胶囊 i记忆之间的余弦相似度， m_i 是幅值，惊讶度 $_i$ 是联想记忆产生的预测误差，而 τ 是温度。

NestedCapsule 在 CIFAR-10 上使用梯度上下文时，准确率达到 78.5 ± 0.6%——比标准 HyperNet 高 $+1.5$ 个百分点，仅比预言机上限（ 79.5% ）低 1.0 个百分点。训练期间，路由熵从 $\log_2 8 = 3.0$ 比特（均匀分布）降至 0.52 比特，表明涌现出了胶囊专门化。胶囊输出之间的方向余弦为 -0.09（近乎正交），证实了功能分化。

6.6 有效秩分析

关于 HyperNet Le 我们计算所有 4 个随机种子下 V 矩阵的有效秩 s:

Seed	r_{eff}	名义秩	利用率	$C_{\text{ctx}}^{\text{rank}}$（比特， $b = 1$）
0	59.8	64	93.4%	59.8
1	58.9	64	92.0%	58.9
2	59.7	64	93.3%	59.7
3	59.7	64	93.3%	59.7
Mean	59.5 ± 0.4	64	93.0%	59.5 ± 0.4

有效秩为 $r_{\text{eff}} \approx 59.5$ （满秩 64），表明上下文通道使用了其名义容量的 93%。上下文信道容量（以比特为单位）为：

$$C_{\text{ctx}}^{\text{rank}} \approx 59.5 \times b \gg H(T) = \log_2 5 \approx 2.3 \text{ bits} \tag{46}$$

即使按最保守的估算，每个维度仅有 $b = 1$ 比特，容量也超过 H（T） $\sim 26\times$ 倍。这种严重的过度配置解释了 HyperNet 为何能实现完美的任务分离：上下文通道的容量远超所需，因此任务识别问题可以轻而易举地解决。

对扩展的启示。如果我们需要支持 K 个任务，并且 H（T） $=$ 对数 $_2 K$，所需的秩仅按 O（对数 K）扩展。秩为 64 且 $b = 1$ 时，原则上我们可以支持 $2^{59.5} \approx 8.5 \times 10^{17}$ 个任务——上下文通道并非扩展的瓶颈。

7 讨论

7.1 架构 $>$ 算法：C_{ctx}优先的设计原则

我们在 1,130+ 个实验中得到的核心启示是，信息流的拓扑结构远比在该拓扑结构中运行的算法重要。EWC、突触智能和 LwF 都采用了复杂的正则化算法，但在 Split-MNIST 上的准确率仅为 $\leq 24.2\%$——对于后续任务来说，仅略高于随机猜测水平。与此同时，一个上下文编码器极其简单的 HyperNetwork（将独热任务标识输入单层多层感知机）实现了 98.8% 的准确率，且没有遗忘。差异完全源于结构。

我们将其提炼为持续学习架构的三项具体设计原则：

1. **明确的上下文信号**。 架构必须有一个定义明确的输入，用于承载任务识别信息——无论是任务身份、批次统计量还是梯度特征。没有这一信号，$C_{\text{ctx}} = 0$ 按定义就无法实现，任何算法都无法防止遗忘（推论 2）。

2. **结构上不可绕过性**。 所有特定于任务的计算都必须经由上下文通路进行。不能存在绕过上下文即可编码任务信息的参数通路。形式化地，对于参数生成器 $\theta(c) = g(c, \phi)$，其中 ϕ 表示上下文无关参数，我们要求$I(\theta; T \mid c) = 0$——参数所携带的任务信息不能超出上下文提供的信息。

3. **可微上下文编码**。 从原始观测到上下文向量的映射必须相对于编码器参数端到端可微。这不只是工程上的便利：我们的学习式编码器实验表明，基于 EMA 的运行统计量（$\bar{\mu}_k, \bar{\sigma}_k^2$）使用了自动微分无法追踪的原地缓冲区操作，导致编码器退化为近乎恒定的映射（$\Delta_{\text{P5}} = 0$）。将 EMA 替换为直接批次统计量（direct_stats=True）后，Split-MNIST 上的结果从 0.0 恢复至 -92.2 个百分点。

CFlow（NeuralDrift团队，2026）生动地展示了违反原则 2 会发生什么。其架构将一个 32 维的上下文向量与一个 4,842 维的参数向量 θ 拼接起来，作为常微分方程流网络的输入。尽管上下文编码器定义良好（原则 1）且可微（原则 3），θ 与 c 之间 150:1 的维度比仍会造成结构性绕过：对优化器而言，将所有任务信息编码到元学习初始化 θ_0中，比通过狭窄的上下文通路传递梯度在能量上代价更低。结果就是 $\Delta_{\text{P5}} = 0.0$——模型完全忽略了上下文信号——所有性能（准确率 $= 92.4\%$）都来自记忆下来的 θ_0。

备注 4. C_{ctx}优先原则颠倒了典型的持续学习研究流程。我们不该问“给定固定架构，什么算法可以防止遗忘？”，而应问“为了满足 $C_{\text{ctx}} \geq H(T)$的需求，什么架构能确保上下文通路无法被绕过？”算法是次要的。

7.2 学习式上下文编码器问题

实际部署中的一个关键问题是：上下文信号能否从数据中学习，而非由预言机提供？我们的实验表明，答案很大程度上取决于基准。

MNIST：批次统计量已足够。在 Split-MNIST 上，5 个任务涉及不同的数字对（0/1、2/3、.....、8/9），它们的像素级统计量存在显著差异。不同任务的批次均值向量之间的两两余弦相似度约为 0.5，为任务判别提供了充足信号。采用直接批次统计量的学习式编码器达到 99.1% 的准确率（$\Delta_{\text{P5}} = -95.2$），实际上超过了预言机编码器（98.8%）。

CIFAR-10：批次统计量失效。在 Split-CIFAR-10 上（5 个任务，每个任务 2 个类别），所有自然图像类别的低层统计量都相似。批次均值之间的两两余弦相似度超过 0.995，而学习式批次统计编码器的准确率仅为 54.4%——相当于只记住了最后一个任务。

解决方案：梯度上下文编码器。为解决这一问题，我们提出使用损失梯度 $\nabla_{\theta_{\text{base}}} \mathcal{L}$ 作为上下文信号。关键在于，梯度天然具有任务判别性：即使输入统计量无法区分，不同任务也会在共享参数上诱导出不同的更新方向。具体而言，我们计算一个小批次梯度，通过冻结随机矩阵将其投影到 $\mathbb{R}^{128}$，再将结果输入一个小型多层感知机，生成一个 64 维的上下文向量。

梯度上下文编码器大幅缩小了预言机差距：

$$\text{Batch statistics:} \quad \text{ACC} = 54.4\%, \quad \Delta_{\text{oracle}} = 23.3\text{pp} \tag{47}$$

$$\text{Gradient context:} \quad \text{ACC} = 77.0 \pm 0.9\%, \quad \Delta_{\text{oracle}} = 0.7\text{pp} \tag{48}$$

对任务间梯度余弦相似度的分析证实了这一机制：使用真实标签计算的梯度，其成对余弦相似度为 ≈ -0.19 （不同任务间近乎正交）；而伪标签梯度的余弦相似度为≈ 0.95 （无法区分任务）。因此，梯度上下文编码器的有效性，在很大程度上取决于计算上下文时能否获取真实标签。

局限性。 梯度上下文编码器需要在推理时使用带标签的样本来计算 $\nabla_{\theta_{\text{base}}}\mathcal{L}$。在部署场景中，这意味着系统必须为每个任务准备一个小型带标签校准集——类似于"任务提示"——才能生成适合任务的参数。虽然在许多实际场景中（例如少样本校准）这可以接受，但它无法实现完全无监督部署。

7.3 "冻结 > 学习式"现象

在我们的实验活动中，一个反复出现、起初令人费解的发现是：冻结的随机组件往往优于其学习式对应组件：

表 7：例 NeuralDrift 项目中观察到的"冻结 > 学习式"现象实例 t.

组件	冻结准确率	学习式准确率	Δ
DND 特征提取器	81.95%	80.85%	+1.10
SPC-TC 词典	≈ 学习式	≈ 冻结式	≈ 0.0
梯度投影矩阵	77.0%	—	（无学习式变体）

该 C_{ctx} 框架提供了两种互补的解释：

容量过剩。 当特征空间 $C_{\text{comb}} = \log_2 \binom{N}{k}$ 的组合容量远远超过标签熵 $H(Y)$ 时——过参数化设置中通常正是如此——特征的具体选择对分类准确率影响甚微。真正重要的是特征空间在不同任务间的稳定性 。冻结随机特征具有完美的稳定性（按构造漂移为零），而学习特征在针对新任务训练时不可避免地发生漂移，可能会损害旧任务的表征。

元学习旁路塌缩。 在可学习组件 h_ψ与高容量静态路径（例如 θ_{base} 在 HyperNetwork 中）并存的元学习架构中，优化器面临一个选择：是投入 h_ψ ，还是在 θ_{base}中编码相同的信息。由于 θ_{base} 通过标准反向传播更新，而 h_ψ 需要二阶梯度（通过上下文到参数的映射），阻力最小的路径就是让 h_ψ 塌缩为近似常数函数，并将所有与任务相关的信息编码到 θ_{base}中。

命题 11 (元学习旁路塌缩)。 考虑一个在任务 $\{D_k\}$上进行优化的参数生成器 $\theta_k = \theta_{\text{base}} + V \cdot h_\psi(c_k)$。如果 ***(i)*** θ_{base} 具有足够的容量来拟合任何单个任务，且 ***(ii)*** 对 θ_{base} 进行梯度更新的成本低于通过 h_ψ进行二阶更新的成本，那么优化器将收敛到 $h_\psi(c_k) \approx \text{const}$，适用于所有 k——这是一个忽略上下文的不动点。

该命题解释了为何朴素的 CFlow 架构会失败（$\Delta_{\text{P5}} = 0$），也解释了为何必须采用明确的机制来强制 模型依赖上下文——例如基础参数丢弃（在训练期间随机将 θ_{base} 置零）或对比式上下文损失——以打破旁路均衡。

7.4 对持续学习研究的启示

负面结果的价值。我们的实验活动产生了 $15+$ 个已关闭的研究方向，每个方向都记录了明确的数学或实证根本原因。这些并非超参数调优失败，而是在给定架构类别中存在的结构性不可能。例如，赫布学习对 DND 的持续学习性能贡献为 0.0 个百分点，因为冻结的随机投影能够达到相同的准确率——赫布规则简化为在线主成分分析，恰好舍弃了任务判别所需的高频成分。我们认为，持续学习社区若能系统地

发表此类负面结果，将大有裨益，因为它们缩小搜索空间的效率，远高于正面结果拓展搜索空间的效率。

生物学启发需要生物尺度。 我们放弃的几个方向——赫布学习、代谢剪枝、突触巩固——都受到神经科学的启发。在大脑中，这些机制作用于 10^{11} 个神经元，并受到复杂的神经调制控制（多巴胺、乙酰胆碱、去甲肾上腺素）。当将赫布学习简化为采用 Oja规则的两层MLP 时，它便退化为主成分分析——一种线性降维方法，既不针对特定任务，也无法与随机投影相媲美。由此得到的启示并不是生物机制无关紧要，而是它们需要达到生物尺度的复杂程度才能发挥作用。

Split-MNIST 是一个危险的基准。 我们的结果表明，Split-MNIST 的任务间差异非常大，以至于几乎任何上下文信号都足以奏效：仅批次像素均值就能达到 99.1% 的准确率。这意味着，Split-MNIST 无法区分具有高 C_{ctx} 和具有低但非零 C_{ctx}的架构。在任务间差异较为细微的 CIFAR-10 上，预言机（77.7%）与朴素的学习编码器（54.4%）之间相差 23.3 个百分点——这种失效模式在 MNIST 上完全无从察觉。我们建议，持续学习论文始终至少纳入一个“困难上下文”基准，在该基准中，批次统计量无法提供有效信息。

评估方法。 有两项方法论发现值得关注：(i) 快速模式评估（减少训练轮次和样本）在我们的持续学习实验中产生了 100% 的假阳性率——所有在快速模式下看似有效的方法，在全面规模的实验中都失败了。我们不建议在持续学习评估中采用缩短的训练协议。(ii) 错误上下文探测（P5）应成为任何上下文条件持续学习架构的标准评估工具，因为它能直接、清晰地衡量架构是否真正利用了其上下文信号。

7.5 局限性

基准测试范围。 我们的系统性验证在 Split-MNIST（5 个任务）上进行，并在 Split-CIFAR-10 上开展了扩展实验。在更具挑战性的基准测试——Split-CIFAR-100（20 个任务）、Split-ImageNet 和域增量设置——上验证 C_{ctx} 预测，仍是未来一项重要工作。关键问题在于，随着任务数量增加、任务间相似度提高，明显的 $C_{\mathrm{ctx}} = 0 \Leftrightarrow$ 遗忘二分现象是否依然存在。

任务增量假设。该 C_{ctx} 框架假设存在能够承载任务识别信息的上下文信号，或能够推断出此类信号。在无任务持续学习设置中，任务边界未提供且数据分布逐渐偏移，“上下文”的概念不太明确。将框架扩展到连续分布偏移的情形并非易事；在这种情形下， T 是连续随机变量，而非离散的任务索引。

将 P5 作为二元探测指标。错误上下文探测协议提供的是一个二元信号：模型要么使用上下文（$\Delta_{\mathrm{P5}} \ll 0$），要么不使用（$\Delta_{\mathrm{P5}} \approx 0$）。它无法对 C_{ctx} （以比特为单位）提供连续且经过校准的估计。使用 $\Delta_{\mathrm{P5}} = -95\%$的方法和使用 $\Delta_{\mathrm{P5}} = -50\%$的方法都“使用上下文”，但上下文通路承载的任务信息量有所不同。开发连续估计器（例如通过 MINE（Belghazi等人，2018）或 NWJ界）将有助于完善该框架。

任务边界知识。 超网络架构要求在训练期间掌握任务边界（以累积每任务上下文统计量并应用对比损失）。虽然这是任务增量持续学习中的标准假设，但它限制了该方法在任务转换可识别的场景中的适用性。

梯度上下文的标签要求。 梯度上下文编码器需要真实标签来计算判别性梯度。伪标签梯度（跨任务的余弦 ≈ 0.95 ）不含有用信息。这限制了完全无监督部署，不过在许多实际场景中，少样本校准是可行的。

7.6 未来方向

扩展到更难的基准测试。当务之急是在 Split-CIFAR-100（20 个任务，每个任务 5 个类别）和 Split-ImageNet 上验证 C_{ctx} 框架，其中 $H(T) = \log_2 20 \approx 4.3$ 比特要求更高容量的上下文通道。梯度上下文编码器是自然的候选方案，因为基于梯度的任务判别应随模型容量扩展。

通过自动上下文生成实现无任务持续学习。NestedCapsule架构（第6.5节）提供了一个有前景的方向：其路由机制在没有显式任务标签的情况下产生涌现式任务特化（门控熵从 $\log_2 8 = 3.0$ 降至 ~ 0.5，方向余弦为 ≈ -0.09）。这表明一条通向无任务持续学习的路径，其中上下文由输入统计量和路由动态自动生成。

连续 C_{ctx} 估计。将二元P5探针替换为基于互信息神经估计（Belghazi等人，2018年）或Nguyen-Wainwright-Jordan界的连续估计器，可以定量比较那些都"使用上下文"、但使用程度各不相同的架构。这对于比较HyperNet的不同变体（预言机、学习式或梯度式）尤其重要。

与PAC-Bayes界的联系。 C_{ctx} 框架与持续学习的PAC-Bayes分析（Doan 等人，2021年）有着天然联系：上下文信道容量发挥着复杂度项的作用，用以控制泛化-遗忘权衡。将这种联系形式化，或可得出更紧的上界，并将上下文条件架构的特定结构纳入考量。

多模态与组合式上下文。现实世界中的持续学习通常涉及多个同时存在的上下文信号（任务指令、环境线索、用户身份）。将 C_{ctx} 扩展到多通道设置，其中 $C_{\text{ctx}} = I$（c_1、 c_2、……、 c_m； θ），可以为组合式持续学习系统提供设计指导。

8 相关工作

8.1 灾难性遗忘

灾难性遗忘问题最早由McCloskey和Cohen（1989）在联结主义模型中发现，并由French（1999）进行了系统综述。现代持续学习研究由弹性权重巩固（EWC）（Kirkpatrick等人，2017）催化，该方法利用Fisher信息矩阵惩罚对过去任务中重要参数的修改。突触智能（SI）（Zenke等人，2017）在训练过程中在线计算重要性，从而无需单独计算Fisher信息。无遗忘学习（LwF）（Li和Hoiem，2017）利用先前模型输出进行知识蒸馏，无需存储数据。经验回放及其变体（Chaudhry等人，2019）会存储并重新访问部分历史数据。

从 C_{ctx} 的角度来看，上述方法都属于 $C_{\text{ctx}} = 0$ 方法：它们作用于共享参数，却没有任何用于任务条件计算的机制。我们的框架预测——实验也证实——无论其正则化或重放策略多么复杂，这类方法都存在根本性局限。

8.2 基于架构的持续学习

渐进式网络（Rusu等人，2016）为每个任务分配新的网络列，以参数数量线性增长为代价实现零遗忘。PackNet（Mallya和Lazebnik，2018）通过迭代剪枝将参数子集分配给各个任务。Expert Gate（Aljundi等人，2017）训练特定于任务的专家网络，并采用门控机制进行任务选择。

这些方法通过提供特定于任务的参数通路，隐式地实现了 $C_{\text{ctx}} > 0$，但通常需要与 K成正比地增加参数数量。HyperNetwork 方法（von Oswald 等人，2020）实现了更优雅的解决方案：固定大小的元网络从紧凑的上下文向量生成特定于任务的参数，使总参数数量不受 K影响。

von Oswald 等人（2020）首次证明，任务条件超网络在 Split-MNIST 和 Permuted-MNIST 上可以取得出色的持续学习性能。Ehret 等人（2021）将这一方法扩展到采用学习式任务嵌入的序列任务设置。相较于这条研究脉络，我们的贡献并非架构方面的——我们使用的是标准超网络——而是解释性的： C_{ctx} 框架提供了超网络为何成功的正式信息论解释，指出了必不可少的精确结构性质（不可绕过性、可微性），并预测了若无此框架便无法察觉的失效模式（CFlow 的上下文绕过）。

8.3 持续学习的信息论视角

此前已有多项研究将信息论应用于持续学习，但没有一项分析上下文条件架构的作用。

Achille 和 Soatto（2018）证明，权重与训练数据之间的互信息 $I(W;S)$ 通过 PAC-贝叶斯式界控制泛化。然而，他们的分析仅限于单任务学习，并未涉及会产生遗忘的序贯多任务设定。

Doan 等（2021）通过神经正切核（NTK）重叠矩阵分析灾难性遗忘，并为持续学习推导了 PAC-Bayes 界。他们的分析从任务梯度相似性的角度刻画遗忘，但未考虑参数由上下文条件生成的架构。

Taheri 和 Thrampoulidis（2025）推导了过参数化单隐层网络的遗忘率界，表明遗忘与任务专属特征相对于共享特征的比例成正比。这些上界适用于 $C_{\mathrm{ctx}} = 0$ 架构，未考虑条件参数生成。

BEDS（Caraffa，2026）指出，在对分布偏移进行贝叶斯估计时，遗忘在结构上不可避免，并提出了 Fisher-Rao 最优正则化。尽管其理念与我们的不可能性结果（定理 3）一致，但 BEDS 提供的是宏观框架，缺少使我们的工作脱颖而出的架构分析和定量 C_{ctx} 预测。

我们的框架的关键区别在于关注架构的信息流拓扑：我们分析的不只是遗忘是否发生，还分析信息经由上下文通路的具体路由方式如何决定遗忘是否必然发生。这使问题从“遗忘了多少？”转变为“该架构是否允许实现零遗忘？”

8.4 压缩与持续学习

压缩与学习之间的联系由来已久，从最小描述长度（MDL）（Rissanen，1978）到信息瓶颈（Tishby 等人，2000）。在持续学习的背景下，Bornschein 等人（2022）将序贯 MDL 与在线学习联系起来，指出数据流的可压缩性与可学习性相关。Angelini 等人（2023）将 MDL 最直接地应用于持续学习，采用基于 MDL 目标的变分密度传播。他们的研究表明，基于 MDL 的持续学习是可行的，但并未分析架构结构在决定压缩效率方面所起的作用。

Finzi 等人（2025）提出超复杂性，作为面向计算能力受限智能体的信息度量，将香农信息扩展到解码器计算能力有限的情形。这与我们的框架相关，因为 C_{ctx} 隐含地假设参数生成器 g 能够将上下文解码为有用的参数——这一假设对计算受限的生成器可能并不成立。

我们的工作与压缩与持续学习文献的不同之处在于，我们并不提出基于压缩的学习算法。相反，我们使用信息论分析来刻画条件再生的架构容量——即从紧凑上下文信号重建任务特定参数的能力。

8.5 元学习与任务条件化

超网络方法在持续学习中的应用与元学习密切相关（Finn 等人，2017；Hospedales 等人，2022），其中元学习器根据任务描述符生成任务特定模型。模型无关元学习（Finn 等人，2017）及其变体学习一个初始化 θ_0，由此可通过少量梯度步获得任务特定参数。在我们的框架中，模型无关元学习的内循环充当隐式上下文编码器：支持集+ 梯度步产生任务条件参数更新。然而，由于 θ_0 在各任务间持续存在并承载模型的大部分容量，模型无关元学习表现出与 CFlow 相同的旁路漏洞——“任务信息”可能主要存在于 θ_0 之中，而非存在于适应过程中。

多任务学习中的任务条件架构（Strezoski 等人，2019）使用 FiLM 层（Perez 等人，2018）、任务特定批归一化或适配器模块来调制共享主干网络。这些方法从设计上实现了 $C_{\mathrm{ctx}} > 0$，因为任务信号会直接调制中间表示。我们的框架为评估此类设计提供了一个有原则的视角：关键在于调制路径是否具有足够的容量（$C_{\mathrm{ctx}} \geq H(T)$），以及是否确实无法绕过。

表8：理论持续学习框架对比。“架构分析” =分析架构拓扑如何影响遗忘。“C_{ctx}” = 明确对上下文信道容量进行建模。“负面结果” =提供系统性的负面结果及其根本原因。

框架	信息论	架构分析	C_{ctx}	实证	负面结果
Achille 和 Soatto（2018）	✓	–	–	✓	–
Doan 等（2021）	✓	–	–	✓	–
Taheri 和 Thrampoulidis（2025）	✓	–	–	–	–
Caraffa（2026）	✓	–	–	✓	–
Angelini 等人（2023）	✓	–	–	✓	–
Ours	✓	✓	✓	✓	✓

8.6 我们在上下文中的贡献

据我们所知，此前没有任何工作同时做到以下几点：(i) 提出与架构上下文容量相关的遗忘正式信息论下界；(ii) 从信息流拓扑而非算法复杂度的角度比较持续学习方法；(iii) 将这样一项原则形式化：从上下文进行条件再生可以绕过不可能三角；以及 (iv) 系统性地给出负面结果（$15+$ 已封闭的方向），并明确其精确的数学或实证根本原因。

表8总结了我们的框架与持续学习领域既有理论工作的关系。

9 结论

我们提出了上下文信道容量（C_{ctx}），这是一个信息论框架，为持续学习中的灾难性遗忘提供了统一解释。从 C_{ctx}的视角来看，持续学习方法的多样图景——从正则化（EWC、突触智能）到蒸馏（LwF）、重放，再到基于架构的方法（超网络）——都归结为一个结构性问题：有多少任务识别信息能够通过架构的上下文通路？

我们的框架有三项贡献：

理论方面。 我们证明，零遗忘需要 $C_{\text{ctx}} \geq H(T)$（定理 4），并证明可通过条件再生绕过不可能三角（定理 3）——即从紧凑上下文信号中重建任务特定参数，而非通过序贯更新来维持这些参数。这些结果将持续学习的问题从“如何防止参数干扰”转变为“如何确保足够的上下文带宽”。

实证方面。 我们在 Split-MNIST 上对 8 种持续学习方法进行了验证（$1{,}130+$ 项实验，每种配置使用 4个随机种子）。 C_{ctx} 代理指标能够完美预测遗忘行为：所有具有 $C_{\text{ctx}} = 0$的方法都会出现灾难性遗忘（6–97%），而具有 $C_{\text{ctx}} \approx 1$ 的方法则实现零遗忘。我们进一步证明，该框架可扩展至更具挑战性的基准：梯度上下文编码器在 Split-CIFAR-10 上达到 77.0% 的准确率（与预言机相差 0.7 个百分点以内）；NestedCapsule架构则将准确率提升至 78.5%，并涌现出路由专门化。

方法论。 错误上下文探测（P5）提供了一种简单实用的诊断工具：输入错误任务上下文，并测量准确率下降。无论架构看起来多么复杂， $\Delta_{\text{P5}} \approx 0$ 都能明确诊断出上下文盲视。我们建议将 P5 作为任何上下文条件持续学习系统的标准评估工具。

总体设计原则是：架构优先于算法，上下文通路必须在结构上不可绕过。无论是 Fisher 正则化、突触智能、知识蒸馏还是赫布学习，再高的算法复杂度也无法弥补这样的架构缺陷：任务识别信息没有影响计算过程的通路。反之，即使是简单的架构，只要上下文通道设计得当，也能实现零遗忘。

展望未来，最重要的开放问题包括：(i) 将 C_{ctx} 扩展到任务数量众多（$K \gg 5$）且任务间相似度细粒度的情形；(ii) 将该框架拓展至无任务持续学习，在这种情况下，必须在不了解任务边界的前提下自动生成上下文；(iii) 将 C_{ctx} 与 PAC-贝叶斯泛化界联系起来，从而为上下文条件持续学习架构提供端到端的学习保证。

参考文献

Alessandro Achille 和 Stefano Soatto。深度表征中不变性与解耦的涌现。机器学习研究杂志，19(50):1–34，2018。Rahaf Aljundi、Punarjay Chakravarty 和 Tinne Tuytelaars。专家门控：利用专家网络实现终身学习。载于*IEEE*计算机视觉与模式识别会议，第3366–3375页，2017。Matteo Angelini 等人。采用描述长度先验的变分持续学习。*arXiv*预印本，2023。Mohamed Ishmael Belghazi、Aristide Barber、Sai Rajeswar、Gabriel Loaiza-Ganem、Saibal Kahou 和 Aaron Courville。MINE：互信息神经估计。载于国际机器学习会议，第531–540页。PMLR，2018。Jörg Bornschein 等人。通过短期学习与增量学习实现神经组合计算。*arXiv*预印本 *arXiv:2209.01975*，2022。Thomas Caraffa。BEDS：持续学习中分布偏移的贝叶斯估计。*arXiv*预印本，2026。Arslan Chaudhry、Marcus Rohrbach、Mohamed Elhoseiny、Thalaiyasingam Ajber、Davide Alessio 和 Martin Jaggi。关于持续学习中的微型情景记忆。载于*arXiv*预印本 *arXiv:1902.10486*，2019。Thomas M Cover 和 Joy A Thomas。信息论基础。John Wiley & Sons，第2版，2006。Thang Doan、Mehdi Abbana Bennani、Bogdan Mazoure、Guillaume Rabusseau 和 Pierre Alquier。通过 NTK 重叠矩阵对灾难性遗忘进行理论分析。载于国际人工智能与统计会议，第1072–1080页。PMLR，2021。Benjamin Ehret、Christian Henning、Maria R Cervera、Alexander Meulemans、Johannes von Oswald 和 Benjamin F Grewe。循环神经网络中的持续学习。载于国际学习表征会议，2021。Chelsea Finn、Pieter Abbeel 和 Sergey Levine。用于深度网络快速适应的模型无关元学习。载于国际机器学习会议，第1126–1135页。PMLR，2017。Marc Finzi、Gregory Benton 和 Andrew Gordon Wilson。超复杂性：香农理论之外复杂系统中的信息。*arXiv*预印本 *arXiv:2601.03220*，2025。Robert M French。联结主义网络中的灾难性遗忘。认知科学趋势，3(4): 128–135，1999。Timothy Hospedales、Antreas Antoniou、Paul Micaelli 和 Amos Storkey。神经网络中的元学习：综述。*IEEE*模式分析与机器智能汇刊，44(9):5149–5169，2022。James Kirkpatrick、Razvan Pascanu、Neil Rabinowitz、Joel Veness、Guillaume Desjardins、Andrei A Rusu、Kieran Milan、John Quan、Tiago Ramalho、Agnieszka Grabska-Barwinska 等人。克服神经网络中的灾难性遗忘。美国国家科学院院刊，114(13):3521–3526，2017。Zhizhong Li 和 Derek Hoiem。无遗忘学习。载于欧洲计算机视觉会议，第614–629页。施普林格，2017。Arun Mallya 和 Svetlana Lazebnik。PackNet：通过迭代剪枝向单个网络添加多个任务。载于*IEEE*计算机视觉与模式识别会议，第7765–7773页，2018。Michael McCloskey 和 Neal J Cohen。联结主义网络中的灾难性干扰：序列学习问题。学习与动机心理学，24:109–165，1989。NeuralDrift团队。面向持续学习的连续时间参数流，2026。作为NeuralDrift项目的一部分开发。

Erkki Oja。将简化神经元模型用作主成分分析器。数学生物学杂志，15 (3):267–273，1982。

Ethan Perez、Florian Strub、Harm de Vries、Vincent Dumoulin 和 Aaron Courville。FiLM：使用通用条件层进行视觉推理。载于*AAAI*人工智能会议，2018。

Jorma Rissanen。通过最短数据描述进行建模。$Automatica$，14(5):465–471，197 8.

Andrei A Rusu、Neil C Rabinowitz、Guillaume Desjardins、Hubert Sober、Koray Kavukcuoglu 和 Raia Hadsell。渐进式神经网络。载于*arXiv*预印本 *arXiv:1606.04671*，2016。

Gjorgji Strezoski、Nanne van Noord 和 Marcel Worring。通过任务路由进行多任务学习。载于*IEEE*国际计算机视觉会议，第1375–1384页，2019。

Hossein Taheri 和 Christos Thrampoulidis。关于过参数化持续学习中的遗忘速率。*arXiv* 预印本 *arXiv:2501.xxxxx*，2025。

Naftali Tishby、Fernando C Pereira 和 William Bialek。信息瓶颈方法。*arXiv* 预印本 *physics/0004057*，2000。

Johannes von Oswald、Christian Henning、João Sacramento 和 Benjamin F Grewe。使用超网络进行持续学习。发表于国际学习表征会议，2020。

Friedemann Zenke、Ben Poole 和 Surya Ganguli。通过突触智能实现持续学习。发表于国际机器学习会议，第 3987–3995 页。PMLR，2017。

A 完整证明

A.1 定理 1 的证明（信息瓶颈链）

我们给出完整证明，并明确列出所有条件。

证明。 考虑序贯学习过程 $\theta_0 \to \theta_1 \to \cdots \to \theta_K$，其中 $\theta_k = U(\theta_{k-1},\ D_k)$ 是确定性更新规则。

步骤 1：马尔可夫链条件。 我们声称 $D_1 \to \theta_1 \to \theta_2 \to \cdots \to \theta_K$ 构成一个马尔可夫链。为此，注意到根据序贯学习的因果约束，$\theta_k = U(\theta_{k-1},\ D_k)$ 仅取决于 θ_{k-1} 和 D_k。由于 D_k 是从 D_1 中独立地抽取的（任务彼此独立），因此有：

$$P(\theta_k \mid \theta_{k-1}, D_1, \ldots, D_{k-1}) = P(\theta_k \mid \theta_{k-1}) \tag{49}$$

其中，以 θ_{k-1} 为条件隐含地包括了 D_k 的影响（它与 D_1 独立）。更准确地说，对于 $k \geq 2$：

$$I(\theta_k; D_1 \mid \theta_{k-1}) = 0 \tag{50}$$

因为 θ_k 是 θ_{k-1} 和 D_k 的确定性函数，而且 $D_k \perp D_1$。

步骤 2：数据处理不等式。 数据处理不等式（Cover 和 Thomas，2006）指出，对于任意马尔可夫链 $X \to Y \to Z$：

$$I(X; Z) \leq I(X; Y) \tag{51}$$

应用于链 $D_1 \to \theta_{k-1} \to \theta_k$（这对所有 $k \geq 2$ 都成立）：

$$I(\theta_k; D_1) \leq I(\theta_{k-1}; D_1) \qquad \forall\, k \geq 2 \tag{52}$$

通过归纳，我们得到完整的链条：

$$I(\theta_K; D_1) \leq I(\theta_{K-1}; D_1) \leq \cdots \leq I(\theta_1; D_1) \tag{53}$$

步骤 3：初始容量界。参数 $\theta_1 = U(\theta_0,\ D_1)$ 的取值范围为 $\Theta \subseteq \mathbb{R}^d$。如果每个参数都以精度 δ 表示（例如，浮点量化），那么 $|\Theta| \leq (1/\delta)^d$，并且：

$$I(\theta_1; D_1) \leq H(\theta_1) \leq \log_2 |\Theta| \leq d \cdot \log_2(1/\delta) \triangleq C \tag{54}$$

讨论：参数精度 δ。 </style}对于 IEEE 754 32 位浮点数，尾数有 23 位，因此每个参数的有效精度约为 log() 位（考虑符号位和可区分的尾数状态）。实际上，由于损失景观几何的影响，有效精度往往低得多：参数空间中的大多数方向都是“平坦的”，不包含与任务相关的信息。因此，对于具有 $d \sim 10^6$–10^9 的现代网络，$C = d \cdot 24$位这一上界是一个宽松的上界。

讨论：确定性与随机更新规则。 </style}对于确定性 U（例如，全批量梯度下降），马尔可夫链是确定性的，数据处理不等式显然成立。对于随机 U（例如，带有小批次噪声的随机梯度下降），θ_k 是 $(\theta_{k-1},\ D_k)$ 的随机函数，但马尔可夫性质仍然成立：$I(\theta_k; D_1 \mid \theta_{k-1}) = 0$，因为随机性来自 D_k（与 D_1 独立）和/或随机种子，而非来自 D_1。然而，随机更新只会进一步降低 $I(\theta_k; D_1)$（添加噪声不会增加互信息），因此该上界仍然有效：

$$I(\theta_k; D_1) \leq I(\theta_{k-1}; D_1) - I(\theta_k; D_1 \mid \theta_{k-1}) \leq I(\theta_{k-1}; D_1) \tag{55}$$

其中第二项考虑了随机处理所导致的信息损失 . □

A.2 定理 3 的证明（不可能三角）

We 首先精确定义这三种性质，然后证明它们彼此不相容。

证明。 **形式化定义**。

1. **零遗忘**。对于所有 $j < k \leq K$：

$$\mathrm{ACC}_j(\theta_k) = \mathrm{ACC}_j(\theta_j) \triangleq a_j^* \tag{56}$$

这要求 θ_k 为每个过去任务 j 的预测编码一个充分统计量：存在一个解码函数 ϕ_j，使得对于所有 $k \geq j$，$\phi_j(\theta_k) = f_j^*$（任务 j 的最优预测器）。

2. **在线学习（因果约束）**。步骤 k 的参数仅取决于前一步的参数和当前任务数据：

$$\theta_k = U(\theta_{k-1}, D_k) \tag{57}$$

具体而言，在步骤 k 无法访问 D_1、.....、D_{k-1}（不进行重放）。

3. **有限参数**。参数维度固定：对于所有 k，均为 $\theta_k \in \Theta \subseteq \mathbb{R}^d$，其中 d 与 K 无关。

反证法。假设这三种性质同时成立。根据性质1，θ_K 必须编码关于每个任务 $j \in \{1, \ldots, K\}$ 的充分信息。形式化地，令 $R_j = I(D_j;\ f_j^*)$ 表示在任务 j 上达到准确率 a_j^* 所需的最小互信息。则零遗忘要求：

$$I(\theta_K; D_j) \geq R_j \qquad \forall\, j = 1, \ldots, K \tag{58}$$

如果这些任务彼此独立（即 D_1、.....、D_K 是独立随机变量），那么关于不同任务的信息无法“共享”，因此：

$$\sum_{j=1}^{K} R_j \leq I\left(\theta_K; D_1, \ldots, D_K\right) \leq H(\theta_K) \leq C = d \cdot \log_2(1/\delta) \tag{59}$$

根据性质3，C 是一个与 K无关的常数。但 $\sum_{j=1}^{K} R_j \geq K \cdot R_{\min}$，其中 $R_{\min} = \min_j R_j > 0$（假设每个任务至少需要一定的信息）。对于 $K > C/R_{\min}$，不等式（59）被违反，从而产生矛盾。

超网络为何不违反该定理。超网络生成 $\theta_k = g(c_k; \phi)$，其中c_k 是任务 k 的上下文，而 ϕ 是元学习参数。关键在于，θ_k 并非按序更新的状态——它是由 c_k计算得到的函数值。知识存在于 ϕ中，并在所有任务上联合优化（违反性质 2）。或者，如果 ϕ 在元训练后被固定，那么在测试时，“状态”为(c_k, ϕ)，有效参数数量为 $|\phi| + |c_k|$，其中 $|\phi|$ 可以存储关于所有 K个任务的信息，因为它在训练时能够访问所有这些任务（违反在线约束）。因此，超网络通过重新定义参数生成过程而避开了这一三角形：θ_k 是条件输出，而非累积的状态。

□

A.3 定理 4 的证明（上下文信道容量界）

证明。**设定：将任务识别视为信道编码**。考虑一种上下文条件持续学习架构：在推理时，上下文信号 c 会被映射为参数 $\theta(c) = g(c; \phi)$。我们将其表述为一个通信问题：

- **编码器：**任务身份 $T \in \{1, \dots, K\}$ 会被映射为上下文信号 $c = e(T)$（可能经由学习到的编码器 e）。
- **信道：**上下文信道 $c \mapsto \theta(c)$将信息从上下文传递至模型行为。该信道的容量为 $C_{\text{ctx}} = \max_{P(c)} I(c; \theta(c))$。
- **解码器：**给定参数 $\theta(c)$，模型必须正确分类来自任务 T的输入，这要求模型隐式地恢复任务身份。

步骤 1：应用 Fano 不等式。设 $\hat{T}$ 是基于 $\theta(c)$ 对 T 的估计（“解码后”的任务身份 $\hat{}$）。任务识别的错误概率为 $P_e = P(T \neq T)$。根据 Fano 不等式：

$$H(T \mid \theta(c)) \leq h(P_e) + P_e \cdot \log_2(K - 1) \tag{60}$$

其中 $h(\cdot)$ 是二元熵函数。由于 $H(T \mid \theta(c)) = H(T) - I(T; \theta(c))$，且 $I(T; \theta(c)) \leq I(c; \theta(c)) \leq C_{\text{ctx}}$（根据数据处理不等式，因为 $T \to c \to \theta(c)$），我们得到：

$$H(T) - C_{\text{ctx}} \leq h(P_e) + P_e \cdot \log_2(K - 1) \tag{61}$$

步骤 2：给出错误概率的上界。对于 $K \geq 2$，上述结果给出了 P_e 的下界。使用标准松弛 $h(P_e) \leq 1$ 以及 $\log_2(K - 1) \leq H(T)$：

$$P_e \geq \frac{H(T) - C_{\text{ctx}} - 1}{H(T)} = 1 - \frac{C_{\text{ctx}} + 1}{H(T)} \tag{62}$$

当 K 较大时（即 $H(T) \gg 1$），「$+1$」修正项变得可以忽略：

$$P_e \gtrsim 1 - \frac{C_{\text{ctx}}}{H(T)} \tag{63}$$

步骤 3：将错误概率与遗忘联系起来。当模型未能通过上下文通道识别任务 $\hat{T}$（即 $T \neq T$）时，它会生成针对错误任务优化的参数，从而导致性能下降。我们将其建模为：

$$\overline{\text{Fgt}} \geq P_e \cdot \overline{\text{Fgt}}_{\max} \tag{64}$$

其中 $\overline{\text{Fgt}}_{\max}$ 是使用错误任务的参数时的期望遗忘（可达到的最大遗忘；对于随机猜测水平为 $1/K$ 的 K 类分类，这一值趋近于 $1 - 1/K$）。综合可得：

$$\overline{\text{Fgt}} \geq \max\left(0,\ 1 - \frac{C_{\text{ctx}}}{H(T)}\right) \cdot \overline{\text{Fgt}}_{\max} \tag{65}$$

讨论：Fgt的解释。$\text{Fgt}_{\max}$ 取决于任务结构。对于包含 $K = 5$ 个二分类任务的 Split-MNIST，每个任务内的随机猜测可达到 50% 的准确率，因此

$\overline{\text{Fgt}}_{\max} \approx a^* - 0.5$ 其中 a^* 是可达到的准确率。对于差异很大的任务（例如，数字对 0/1 与 8/9），使用错误任务的参数会使许多任务的准确率接近零，从而得到 $\underline{\text{Fgt}}_{\max} \approx a^*$。在实践中，我们的实验表明 $\text{Fgt}_{\max} \approx 0.97$ （NaiveSGD 遗忘），证实了这一估计。

紧致性。 该上界在两种情况下是紧的：(i) 当 $C_{\text{ctx}} = 0$ 时，我们得到遗忘 $\geq \overline{\text{遗忘}}_{\max}$，$\overline{\text{这可}}$由 NaiveSGD 达到；(ii) 当 $C_{\text{ctx}} \geq H(T)$ 时，该上界给出遗忘 ≥ 0，这可由 HyperNet 达到。 □

A.4 命题证明：Oja 的收敛性与持续学习

命题 12 (Oja 规则与主成分分析收敛性)。 *Oja* 学习规则 $\Delta\mathbf{w} = \eta(\mathbf{x}y - y^2\mathbf{w})$，其中 $y = \mathbf{w}^\top\mathbf{x}$，收敛至数据协方差矩阵 $\mathbf{C} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$ 的最大特征值对应特征向量（即第一主成分方向）。这种收敛性并不能保证持续学习所需的任务判别性表征。

证明。 **步骤 1：Oja 的收敛性**。Oja 更新规则为：

$$\Delta\mathbf{w} = \eta\left(\mathbf{x}y^\top - (y^\top y)\mathbf{w}\right), \qquad y = \mathbf{w}^\top\mathbf{x} \tag{66}$$

代入 $y = \mathbf{w}^\top\mathbf{x}$：

$$\Delta\mathbf{w} = \eta\left(\mathbf{x}\mathbf{x}^\top\mathbf{w} - (\mathbf{w}^\top\mathbf{x}\mathbf{x}^\top\mathbf{w})\mathbf{w}\right) \tag{67}$$

取期望：

$$\mathbb{E}[\Delta\mathbf{w}] = \eta\left(\mathbf{C}\mathbf{w} - (\mathbf{w}^\top\mathbf{C}\mathbf{w})\mathbf{w}\right) \tag{68}$$

这是在约束 $\|\mathbf{w}\| = 1$ 下最大化 $\mathbf{w}^\top\mathbf{C}\mathbf{w}$ 的投影梯度上升。根据瑞利商理论，唯一的最大值（不计符号）是 $\mathbf{v}_1$，即 $\mathbf{C}$ 的最大特征值 λ_1所对应的特征向量。自归一化项 $-(\mathbf{w}^\top\mathbf{C}\mathbf{w})\mathbf{w}$ 确保 $\|\mathbf{w}\|$ 保持有界，收敛性则遵循标准随机逼近结果（Oja，1982）。

步骤 2：主成分分析方向不一定具有任务可判别性。 最大特征向量$\mathbf{v}_1$ 刻画了输入数据中方差最大的方向。然而，最大方差与任务可判别性是不同的性质。考虑均值分别为 $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ 且共享协方差为 $\mathbf{C}_{\text{shared}}$的两个任务。任务可判别方向为 $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$（Fisher 线性判别式），而主成分分析方向是 $\mathbf{C}_{\text{shared}} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$ 的最大特征向量（其中 $\boldsymbol{\mu}$为总体均值）。二者仅当类间方差主导类内方差时才会重合。

步骤 3：稀疏编码的组合容量。对于 $N = 1024$ 个隐藏神经元，采用 top-k激活，且 $k = 80$，可能的激活模式数量为：

$$\binom{N}{k} = \binom{1024}{80} \approx 10^{87} \tag{69}$$

因此，组合容量为：

$$C_{\text{comb}} = \log_2 \binom{1024}{80} \approx 290 \text{ bits} \tag{70}$$

这比任务身份熵 $H(T) = \log_2(5) \approx 2.3$ 比特高出两个数量级以上，表明该表征具有充足的容量。然而，正如我们的 DND 实验（B.1 节）所示，基于 Oja 规则的学习并未利用这种组合容量：任务间实际的神经元重叠为 $> 94\%$（Jaccard指数），这表明 Oja规则会使所有任务趋向相同的稀疏支撑集。

□

A.5 命题证明： S_N 置换对称性

命题 13 (S_N 字典学习的对称性)。 对于重建损失为 $L = \|\mathbf{x} - \mathbf{W}_d\mathbf{z}\|^2$的线性生成模型，同时对$\mathbf{W}_d$ 进行列置换并对$\mathbf{z}$进行相应的置换，损失保持不变。这种对称性会阻碍列专门化的自发形成。

证明。 **步骤 1：置换不变性**。 设 $\pi \in S_N$ 是 $\{1, \ldots, N\}$ 的一个置换，并令 $\mathbf{P}_\pi$ 为相应的 $N \times N$ 置换矩阵。定义：

$$\mathbf{W}'_d = \mathbf{W}_d\mathbf{P}_\pi, \qquad \mathbf{z}' = \mathbf{P}_\pi^\top\mathbf{z} \tag{71}$$

则：

$$L(\mathbf{W}'_d, \mathbf{z}') = \|\mathbf{x} - \mathbf{W}_d \mathbf{P}_\pi \mathbf{P}_\pi^\top \mathbf{z}\|^2 = \|\mathbf{x} - \mathbf{W}_d \mathbf{z}\|^2 = L(\mathbf{W}_d, \mathbf{z}) \tag{72}$$

其中我们使用了置换矩阵的正交性：$\mathbf{P}_\pi \mathbf{P}_\pi^\top = \mathbf{I}$。

步骤 2：对称初始化下的梯度等价性。对于列 i （在 $\mathbf{W}_d$ 中），梯度为：

$$\nabla_{\mathbf{w}_i} L = -2(\mathbf{x} - \mathbf{W}_d \mathbf{z}) z_i \tag{73}$$

在对 $\mathbf{W}_d$ 进行对称（例如独立同分布高斯）初始化的条件下，并假设$\mathbf{z}$ 是作为$\mathbf{W}_d$ 的函数计算的（例如经由 $\mathbf{z} = \mathbf{W}_d^\top \mathbf{x}$ 再经阈值化），则 $(\mathbf{w}_i, z_i)$ 的分布对所有 i 均相同。因此：

$$\mathbb{E}[\nabla_{\mathbf{w}_i} L] = \mathbb{E}[\nabla_{\mathbf{w}_j} L] \qquad \forall\, i, j \tag{74}$$

这意味着，在期望意义上，所有列都接收相同的梯度更新，从而阻止任何列专门化于某一特定任务。

步骤 3：扰动分析。 假设一个小扰动 $\boldsymbol{\epsilon}$ 打破了单列的对称性。重构梯度的幅值为 $\|\nabla_{\mathbf{w}_i} L\| = O(\|\mathbf{x}\|^2)$，它起到“恢复力”的作用，将受扰列拉回对称解。只有当扰动对损失的影响超过该恢复力时，扰动才能增长。对于幅值为 $\|\boldsymbol{\epsilon}\|$ 的扰动，破缺对称梯度分量为 $O(\|\boldsymbol{\epsilon}\| \cdot \|\mathbf{x}\|)$，而每列的重构梯度为 $O(\|\mathbf{x}\|^2/N)$。因此，破缺对称力与重构力之比按 $O(N\|\boldsymbol{\epsilon}\|/\|\mathbf{x}\|)$ 的量级变化，这在高维系统的小扰动下可以忽略。

推论。 任何不会明确地 为不同列分配不同角色的扰动机制（例如生物噪声、随机失活）都会被 $O(\|\mathbf{x}\|^2/N)$ 重构梯度所抑制。这解释了为什么代谢剪枝和时间压力等机制在我们的实验中未能诱导列专门化。

□

A.6 命题证明：元学习旁路塌缩

命题 14 （超网络中的基础参数旁路）。 在一个元学习框架中，当 $\theta_k = \theta_{\text{base}} + \mathbf{V} g_\psi(c_k)$ 以及 θ_{base} 为自由参数时，如果$\mathbf{V}$ 以较小的幅值初始化，则基础路径中的梯度信号会支配上下文路径，从而导致“旁路坍塌”，此时 θ_{base} 会吸收所有任务信息。

证明。 **步骤 1：梯度幅值**。考虑元损失：

$$\mathcal{L}(\theta_{\text{base}}, \mathbf{V}, \psi) = \sum_{k=1}^{K} \mathbb{E}_{(x,y)\sim D_k} \left[\ell\big(f(x;\, \theta_{\text{base}} + \mathbf{V} g_\psi(c_k)),\; y\big) \right] \tag{75}$$

设 $\theta_k = \theta_{\text{base}} + \mathbf{V} g_\psi(c_k)$。相对于各参数组的梯度为：基础路径：

$$\frac{\partial \mathcal{L}}{\partial \theta_{\text{base}}} = \sum_{k=1}^{K} \mathbb{E}\left[\frac{\partial \ell}{\partial \theta_k} \right] \tag{76}$$

这只是各任务梯度之和，没有额外的雅可比因子。

上下文路径（通过 ψ）：

$$\frac{\partial \mathcal{L}}{\partial \psi} = \sum_{k=1}^{K} \mathbb{E}\left[\frac{\partial \ell}{\partial \theta_k} \cdot \mathbf{V} \cdot \frac{\partial g_\psi(c_k)}{\partial \psi} \right] \tag{77}$$

这涉及额外的因子 $\mathbf{V} \cdot \frac{\partial g_\psi}{\partial \psi}$。 —

步骤 2：尺度分析。 在 $\mathbf{V} \sim \mathcal{N}(0, \sigma_V^2)$ 初始化时，其中 $\sigma_V = 0.01$：

$$\left\| \frac{\partial \mathcal{L}}{\partial \theta_{\text{base}}} \right\| = O\left(\left\| \frac{\partial \ell}{\partial \theta} \right\| \right), \qquad \left\| \frac{\partial \mathcal{L}}{\partial \psi} \right\| = O\left(\sigma_V \cdot \left\| \frac{\partial \ell}{\partial \theta} \right\| \cdot \left\| \frac{\partial g}{\partial \psi} \right\| \right) \tag{78}$$

比率为：

$$\frac{\|\partial \mathcal{L}/\partial \psi\|}{\|\partial \mathcal{L}/\partial \theta_{\text{base}}\|} = O\left(\sigma_V \cdot \left\| \frac{\partial g}{\partial \psi} \right\| \right) \approx O(0.01) \tag{79}$$

假设在初始化时 $\|\partial g/\partial\psi\| = O(1)$。

步骤 3：信息吸收。由于基础路径梯度较大（ $\sim 100\times$ ），Adam（或任何自适应优化器）的移动速度都远快于 ψ（ θ_{base} ）。经过几个优化步骤后， θ_{base} 已经吸收了"平均任务"的解，而逐任务残差 ℓ（f（x; θ_{base}） y）很小。这进一步降低了 $\|\partial\ell/\partial\theta\|$，使上下文路径梯度变得更小，形成一个导致旁路坍塌的正反馈循环：g_ψ（c_k） ≈对所有 k均为常数。

步骤 4：将基础参数随机失活作为解决方案。在训练期间以 p 的概率（基础_丢弃_率）设置 $\theta_{\text{base}} \to 0$ ，会在占训练步骤 p 的一部分步骤中移除基础路径，从而迫使：

$$\theta_k = \mathbf{V} g_\psi(c_k) \cdot \frac{\|\theta_{\text{base}}\|}{\|\mathbf{V} g_\psi(c_k)\| + \epsilon} \tag{80}$$

（通过适当的重新缩放来保持输出幅值）。在这些步骤中， $\partial\mathcal{L}/\partial\psi$ 是唯一的梯度通路，迫使上下文编码器学习任务判别性表征。我们的实验确认， $p = 0.3$ 已足够：在 Split-MNIST 上，P5 变化量从 0.0（已坍塌）降至 -96.4 个百分点（完全依赖上下文）。

重要说明：如果上下文编码器使用不可微统计量（例如 EMA 运行均值/方差），仅靠基础参数丢弃是不够的。直接_stats=True 修复方案（直接使用当前批次统计量，这些统计量可微）是必要的前提。 □

B 方法详情

B.1 Deep NeuralDrift（DND）

DND 是一种受生物学启发的持续学习架构，在为期 86 天、包含 1,130+ 个实验的实验系列中进行了探索。我们介绍该架构，并总结促成这一 C_{ctx}框架的诊断结果。

架构。 DND 网络由三层组成：

- **输入层：** 784 个神经元（展平后的 MNIST 图像）。
- **隐藏层 1：** 1024 个神经元，采用 top-k 稀疏激活（$k = 20$）。
- **隐藏层 2：** 512 个神经元，采用 top-k 稀疏激活（$k = 10$）。
- **输出层：** 10 个神经元（每个类别对应一个），采用基于余弦相似度的分类 on.

赫布学习。 使用带奖励调制的 Oja规则更新权重：

$$\Delta w_{ij} = \eta \cdot r \cdot \left(x_i y_j - y_j^2 w_{ij}\right) \tag{81}$$

其中 r i 是奖励信号（正确时为 1，错误时为 0，"reward_hebbian_neg_only"变体 t).

髓鞘化。 频繁激活的连接会被"髓鞘化"（冻结），由以下参数控制：

- 髓鞘化_阈值：140（激活计数阈值）
- 髓鞘化_缩放：20（sigmoid 门的锐度）

模板边界模板刷新（TBTR）。 在任务边界处，旧任务的测试数据会再次前向通过当前网络，更新分类器模板（EMA 质心），而不修改隐藏层权重。这纠正了漂移特征与过时模板之间的失同步。

诊断实验结果。 表9总结了揭示 DND 架构局限性并促成 C_{ctx} 框架的关键诊断实验。

关键发现是，冻结的随机投影的表现与赫布学习相当，甚至更优（71.98% 对比 72.08%）。结合神经元重叠分析（C.4 节），这表明 DND 的架构$C_{\text{ctx}} = 0$：不存在上下文通路，而 TBTR + 余弦最近质心分类就是其全部机制。

表 9：DND 在 Split-MNIST 上的诊断实验（1024/k80 隐藏层配置）。“冻结随机”以冻结的随机投影取代赫布学习；“打乱标签”随机打乱 TBTR 标签；“最终轮”在训练后对所有任务进行模板刷新；“纯随机投影”采用独立的随机投影基线。

实验	准确率（%）	遗忘率（%）	关键发现
TBTR 1024/k80	73.19	-0.42	原始结果
TBTR-训练	72.08	-0.26	训练数据泄漏 $=1$。1个百分点
冻结随机 + TBTR	71.98	-1.48	赫布学习 $=0$贡献
打乱标签 TBTR	33.81	39.07	TBTR 使用真实特征
TBTR + 最终轮	80.85	1.30	训练后修正 $=+8.8$百分点
纯随机投影 +TBTR	81.95	~ 0	比 DND 高 1.1 百分点
无 TBTR	18.53	96.21	灾难性遗忘总量

B.2 CFlow：连续时间参数流

CFlow 是我们用于持续学习的基于常微分方程的方法，其提出基于这样一个假设：连续时间的参数动力学能够实现平滑的任务转换。

架构。

- **任务网络：** ConvTaskNet，包含 4 个滤波器，产生 $\theta_{\text{维}} = 4842$ 参数。
- **流网络：** 一个神经常微分方程 $f(\theta,\ c,\ t)$，用于生成 $d\theta/dt$。输入维度： $\theta_{\text{维}} + c_{\text{维}} + t_{\text{嵌入}}$。
- **上下文编码器：** 学习式上下文编码器，将批次统计量映射到 $c \in \mathbb{R}^{32}$。
- **时间编码：** 用于正弦时间嵌入的 16 个傅里叶频率。

训练。

- 在 80 个包含任务转换的连续时间回合上进行元训练。
- Adam优化器，学习率为 10^{-3}。
- 30% 重放比例，每个过去任务使用 100 个样本。

结果与 θ_0 记忆器问题。 CFlow 在 Split-MNIST 上的准确率为 92.6%（v1），采用 FiLM 条件化时为 95.3%（v2）。然而，探测发现：

- P5（错误上下文）： $\Delta = 0.0$个百分点——模型完全忽略上下文。
- P6（随机 θ_0）：准确率灾难性下降——所有知识都在 θ_0中。
- FiLM架构毫无贡献（v1 $\approx$v2处于噪声范围内）。

根本原因是维度不匹配：将 $c \in \mathbb{R}^{32}$ 与 $\theta \in \mathbb{R}^{4842}$ 拼接会产生150:1的比例。优化器自然会利用占主导地位的 θ 路径，使上下文在结构上不可见。这就是 C_{ctx} 框架所预测的“上下文绕过”失效模式。

B.3 超网络架构

TaskHyperNetwork。 核心架构通过低秩调制生成任务特定参数：

$$\theta_k = \theta_{\text{base}} + \mathbf{V} \cdot g_\psi(c_k) \tag{82}$$

- $\theta_{\text{基础}} \in \mathbb{R}^{4842}$：可学习的基础参数（从 ConvTaskNet 初始化）。
- $\mathbf{V} \in \mathbb{R}^{4842\times 64}$：低秩投影矩阵（$v$_初始化_缩放 $= 0.01$）。
- g_ψ：架构为 $64 \to 256 \to 256 \to 64$ 的多层感知机（采用 SiLU激活函数，并使用层归一化）。

- 上下文维度：64。

上下文编码器。 我们实现了三种上下文编码器：

1. **预言机：** 独热任务标识 $\to$ 线性投影 $\to \mathbb{R}^{64}$。

2. **学习式（批次统计量）：** 输入批次的逐通道均值和方差 $\to$ 多层感知机 $\to \mathbb{R}^{64}$。关键修复：direct_stats=True 直接使用当前批次统计量（可通过自动微分求导），而非指数移动平均运行统计量（不可微的缓冲区操作）。

3. **梯度上下文：** $\nabla_{\theta_{\text{base}}}\ell \to$冻结的随机投影（$\mathbb{R}^{128}$）$\to$ 多层感知机 $\to \mathbb{R}^{64}$。判别性梯度需要真实标签；伪标签梯度会在各任务间产生余弦相似度 > 0.95（无用）。

防坍塌机制。

- **对比权重**（$\lambda = 0.1$）：通过 $\mathcal{L}_{\text{contr}} = \lambda \sum_{i \neq j} \max(0, \cos(c_i, c_j) - \tau)$将不同任务的上下文向量彼此推开。
- **基础参数丢弃**($p = 0.3$)：在训练期间随机将 $\theta_{\text{基础参数}}$置零，迫使模型依赖上下文生成的增量。

训练协议。进行 200 回合联合多任务优化。每回合随机抽取一种任务排列并按顺序训练，同时重放历史任务。使用 Adam优化器，学习率为 $= 10^{-3}$。

B.4 嵌套胶囊超网络

NestedCapsule架构通过将上下文到秩的映射分散到 $N = 8$ 胶囊记忆模块中，对TaskHyperNetwork进行了扩展。

架构。

- $N = 8$ 个胶囊模块，每个模块包含一个 2 层多层感知机：$64 \to 64 \to 8$（SiLU激活函数）。
- 总秩：$N \times 8 = 64$（与 TaskHyperNetwork 相同）。
- 路由：$\alpha_i = \text{softmax}\big(s_i \cdot m_i \cdot (1 + \sigma_i)/\tau\big)$，其中：$-$ $s_i = \cos(\mathbf{d}_i, c)$：胶囊方向与上下文之间的余弦相似度。$-$ $m_i = \exp(\log m_i)$：学习到的幅度（置信度/专业能力）。$-$ σ_i：来自联想记忆预测误差的惊奇信号（受 Titans 启发，与梯度图分离）。$-$ τ：温度参数。
- 输出：$\theta_k = \theta_{\text{基}} + \mathbf{V} \cdot \text{concat}_i(\alpha_i \cdot \text{胶囊}_i(c_k))$。

方向-幅度分解。 每个胶囊通过指数移动平均在上下文空间中跟踪“专长方向”$\mathbf{d}_i \in S^{d-1}$：

$$\mathbf{d}_i \leftarrow \text{normalize}\big(\mu \cdot \mathbf{d}_i + (1 - \mu) \cdot c_k\big), \quad \text{weighted by } \alpha_i \tag{83}$$

其中 $\mu = 0.9$ 为方向动量。

关键结果。

- 带有梯度上下文的 CIFAR-10：78.5% $\pm$ 0.6% 准确率，0% 遗忘（HyperNet 为 77.0%）。
- 门控熵从 2.1（8 个胶囊均匀分布时）降至 0.52，表明路由集中度有所提高。
- 胶囊方向之间的方向余弦：-0.09（反相关），表明存在真正的专门化。
- P6（随机基座）：从 $\sim$50%（HyperNet）降至 10.5%，证实胶囊增量承载了更多模型身份信息。

B.5 基线方法实现

所有无上下文基线均使用一个隐藏层大小为 (256，128) 的 2 层多层感知机、ReLU 激活函数和一个 10 类输出头。训练使用 Adam优化器，学习率为 $=10^{-3}$，批次大小为 64，每个任务训练 5 轮（完整模式）或 2 轮（快速模式）。

- **NaiveSGD：**不使用持续学习机制的标准微调方法，作为遗忘上界。
- **EWC**（Kirkpatrick等人，2017）：每个任务结束后，计算对角Fisher信息矩阵$\mathbf{F}=\mathbb{E}[\nabla\log p(y|x,\theta)\nabla\log p(y|x,\theta)^\top]$。正则化损失：$\mathcal{L}_{\text{EWC}}=\lambda\sum_i F_i(\theta_i-\theta_i^*)^2$，其中 $\lambda=400$。
- **SI**（Zenke等人，2017）：在训练期间跟踪逐参数重要性$\Omega_i=\sum_k \Delta_i^k/(\Delta\theta_i^k)^2$。正则化：$\mathcal{L}_{\text{SI}}=c\sum_i\Omega_i(\theta_i-\theta_i^*)^2$，其中 $c=1.0$。
- **LwF**（Li和Hoiem，2017）：训练前保存教师模型在当前任务数据上的输出。蒸馏损失：$\mathcal{L}_{\text{LwF}}=\alpha\cdot\text{KL}(\text{softmax}(z_T/T)\|\text{softmax}(z_S/T))$，其中 $\alpha=1.0$，温度为 $T=2.0$。
- **经验回放：**为每个过去任务维护一个包含 200 个样本的缓冲区，并在新任务训练期间均匀重放这些样本。不使用正则化。

C 扩展结果

C.1 按随机种子划分的准确率矩阵

我们展示了代表性方法的完整 5×5 准确率矩阵。条目（i，j）表示在任务 i训练后，模型在任务 j 上的准确率。

NaiveSGD（随机种子 0）。

$$\mathbf{A}_{\text{SGD}}=\begin{pmatrix}99.5 & 0.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 98.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 99.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.0 & 98.5 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 99.3\end{pmatrix}$$

每完成一个任务的训练后，只有最近的任务能达到较高的准确率；所有先前任务都遭到灾难性遗忘（准确率为 0%）。这是 $C_{\text{ctx}}=0$ 方法的典型模式。

HyperNet Oracle（随机种子 0）。

$$\mathbf{A}_{\text{HN}}=\begin{pmatrix}100.0 & 97.5 & 98.8 & 99.8 & 97.2\\ 100.0 & 97.5 & 98.8 & 99.8 & 97.2\\ 100.0 & 97.5 & 98.8 & 99.8 & 97.2\\ 100.0 & 97.5 & 98.8 & 99.8 & 97.2\\ 100.0 & 97.5 & 98.8 & 99.8 & 97.2\end{pmatrix}$$

超网络在所有行中均产生相同的准确率，这证实了零遗忘：每个任务的准确率完全由上下文信号决定，而非训练历史。

CFlow（随机种子 0）。

$$\mathbf{A}_{\text{CFlow}}=\begin{pmatrix}97.0 & 90.0 & 93.0 & 94.5 & 92.0\\ 88.5 & 95.0 & 93.0 & 94.0 & 93.5\\ 87.5 & 90.5 & 96.0 & 94.0 & 93.0\\ 87.0 & 89.0 & 92.5 & 96.5 & 93.0\\ 88.5 & 91.0 & 92.0 & 93.0 & 98.5\end{pmatrix}$$

CFlow 表现出中等遗忘：在训练所有任务后，任务1的准确率从 97.0% 降至 88.5%。相对较高的总体准确率（92.4%）源于元学习初始化 θ_0（“θ_0 记忆器”效应），而非上下文相关参数生成。

表10：Permuted-MNIST结果（可用时采用4个随机种子）。Permuted-MNIST比Split-MNIST更难，因为各任务共享同一标签空间，因此任务识别更加关键。

方法	准确率（%）	遗忘率（%）	P5 Δ (pp)
HyperNet 预言机	90.3 ± 0.7	0.0	-75.3
HyperNet 学习式	89.1 ± 0.8	0.0	-73.3
CFlow v1	73.0	—	0.0
随机梯度下降 +重放	66.5	—	0.0

C.2 Permuted-MNIST 结果

表10展示了 re Permuted-MNIST 上的结果（5 个任务，每个任务都是对完整 MNIST 图像像素进行随机置换）.

该 C_{ctx} 框架正确预测了结果：具有较高 C_{ctx} 的 HyperNet 方法实现了零遗忘，而 CFlow 和随机梯度下降+重放（两者均为 $C_{\text{ctx}} = 0$）则表现出显著遗忘。学习到的编码器达到 P5 $\Delta = -73.3$个百分点，证实直接_stats=True 能够推广到 Split-MNIST 之外。

C.3 CIFAR-10 结果

表11展示了 Split-CIFAR-10（5个任务，每个任务包含2个类别）上的结果。该基准揭示了基于批次统计量的上下文编码的局限性。

表11：Split-CIFAR-10 结果（4个随机种子）。不同任务间的批次统计量几乎完全相同（余弦相似度 > 0.995），导致学习到的编码器失效。梯度上下文通过将损失梯度用作上下文信号解决了这一问题。

方法	准确率（%）	P5 Δ(pp)	注释
HyperNet Oracle	77.7	—	独热任务标识
学习型超网络（批次）	54.4	weak	cos > 0.995 跨任务
HyperNet 梯度上下文	77.0 ± 0.9	-77.0	预言机差距仅为 0.7 百分点
NestedCapsule 梯度上下文	78.5 ± 0.6	—	新的 SOTA
NestedCapsule Oracle	79.5 ± 0.9	—	比 HyperNet Oracle 高 +1.8 百分点
随机梯度下降 +重放	∼46–63	0.0	灾难性遗忘

梯度上下文编码器。关键洞见在于，尽管不同 CIFAR-10 划分的批次像素统计量几乎完全相同（余弦相似度 > 0.997），但使用真实标签的损失梯度 $\nabla_{\theta_{\text{base}}} \ell$ 会在不同任务之间产生近乎正交的上下文向量（余弦相似度 $= -0.19$）。梯度上下文编码器将这些梯度投影到一个冻结随机矩阵（$\mathbb{R}^{4842} \to \mathbb{R}^{128}$）中，再通过多层感知机生成 $c \in \mathbb{R}^{64}$。

要求： 梯度上下文编码器需要真实标签来产生判别性梯度。使用伪标签（来自当前模型的预测）时，不同任务之间的梯度余弦相似度为 > 0.95，因此无法用于任务识别。这与 C_{ctx} 框架一致：伪标签梯度携带的是有关模型行为的信息（如果模型尚未专门化，这种行为与任务无关），而非有关任务身份的信息。

C.4 神经元重叠分析（DND）

我们分析 DND 架构中不同任务之间神经元共享的程度，以量化任务专门化的缺失。

指标。**对于每** 组任务对（i， j），我们计算其最活跃的前 k 个神经元集合的 Jaccard 重叠度 s:

$$J(i,j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{84}$$

其中 S_i 是由任务 i 的数据激活的神经元集合。

结果。

• **输出层：** Jaccard 重叠度 $= 0.947$ ——几乎所有任务都共享几乎所有神经元。这意味着，无论输入任务是什么，稀疏输出编码都几乎相同。

• **隐藏层：** $> 60\%$重叠，其中 $< 7\%$的神经元是任务唯一的。

• **模板相似度：**任务间模板余弦相似度（0.751）超过任务内相似度（0.737），这意味着来自不同任务的类别模板彼此之间比同一任务内的模板更为相似。

这些结果证实了A.4节的理论预测：Oja规则会将所有任务推向相同的主成分分析方向，从而阻止任务特定表示的形成。组合容量（ $\binom{1024}{80}$对应$\sim$290 比特）从未被利用。

C.5 ClosedDirection总结

表12全面总结了项目期间已探索并随后关闭的所有研究方向，在1,130+ 次实验中共计 15+ 个方向。

表12：已关闭的研究方向总结。每个方向在关闭前都使用多种超参数配置和随机种子进行了探索。“根本原因”指出了失败的根本原因。

方向	#实验	根本原因
余弦阈值门控-ing	$\sim$20	门控值聚集在阈值附近；二元开/关，没有分级路由
输出层调优 (n/k/η)	$\sim$30	输出层有 $C_{\text{ctx}} = 0$；调优参-数无法创建上下文
髓鞘化扫描	$\sim$25	冻结权重保留当前状态，但无法不创建任务路由
分割/锁定快慢通路	$\sim$15	没有上下文时，快慢通路收敛到相同的表征
超额容量 +超高稀疏	$\sim$20	容量更大 $\neq$ 专门化程度更高，而不会发生对称性破缺
模极排斥	$\sim$15	强度 0.1：没有影响。强度 0.5：准确率下降 4 个百分点（具有破坏性）
标签 嵌入 (FF)	$\sim$10	结构性训练/测试不匹配：训练时有标签，测试时却没有
空间能量约束束	$\sim$20	距离加权 L1 只是一种正则化；在容量压力下有害
权重稀疏性	$\sim$10	68%稀疏性带来 $=$ 0个百分点的准确率增益；稀疏性 $\neq$ 遗忘减少
河床效应	$\sim$15	$E = -\sum a_i \log(\pi_i + \epsilon)$ 提供统一的上限-pression，而非选择性路由
SPC-TC (ISTA 稀疏编码)	$\sim$40	信息瓶颈：类别可分性下-降12.4%。冻结随机字典 $\approx$ 学习式dict
CCD 容量预测-结果	$\sim$15	代数界成立，但容量扩展和收缩阶梯失效
OrthoCorr（梯度校正）	$\sim$15	自适应阈值下的结构冲突；在概念切换时阻断梯度
代谢剪枝（χ-衰减）	$\sim$20	激活频率（$\rho = 0.916$）可区分——但已由自适应机制处理阈值；χ 不携带任何信息（$\rho = 0.018$）
HSPC-T （时间压力）	$\sim$14	10/14 通过，但 $=$动态稀疏性调节器，而非压力 $\rightarrow$专门化

这些失败有一个共同点：缺少上下文通路——所有旨在提升持续学习性能的方向，都试图在 $C_{\text{ctx}} = 0$ 架构内实现这一目标。 C_{ctx} 框架解释了为什么所有这类尝试在根本上都受到限制。

D 实现细节

D.1 计算环境

- **硬件：** 配备支持 CUDA 的图形处理器的英伟达 DGX 工作站。
- **软件：** Python 3.12、PyTorch 2.x、conda 环境“neuraldrift”。
- **实验总数：** 1,130+ 个结果目录，涵盖 86 个开发天数 .
- **代码库：** 134 个文件，共 102,468 行代码。
- **典型运行时间：**

  – 快速模式（1个随机种子，缩减数据）： ∼5 分钟（所有方法）。–
  完整模式（ 4 个随机种子，完整数据）： ∼30 分钟（所有方法）。–
  单次 HyperNet 训练（ 200 回合）：每个随机种子耗时 ∼4 秒。–
  单次 CFlow 训练（ 80 回合）：每个随机种子耗时 ∼20 秒。

D.2 可复现性

随机种子控制。 所有实验均通过共享工具进行确定性设种：

- torch.manual_seed(seed)
- numpy.random.seed(seed)
- random.seed(seed)
- torch.cuda.manual_seed_all(seed)（当 CUDA 可用时）

默认随机种子： $\{0, 1, 2, 3\}$ 用于四种子实验。除非另有说明，所有报告的均值和标准差均基于这些随机种子计算。

结果存储。 每次实验运行都会生成一个带时间戳的目录：

results/{experiment_name}_{YYYYMMDD_HHMMSS}/
results.json # 完整指标、配置、分种子结果 *.png # 图表（准确率矩阵、
曲线） command.txt # 用于复现的确切命令

配置序列化。 所有超参数均以 JSON 可序列化字典的形式存储在 results.json 中，从而支持精确复现。配置包括模型架构、优化器设置、数据预处理和评估协议。

D.3 CCCUnifiedExperiment 脚本

统一对比实验通过单个脚本实现（1,377 行代码）：

experiments/cflow/exp_context_channel_capacity.py

命令行接口。

- --quick：快速模式（2 个训练轮次、每类 200 个训练样本、减少元训练回合数）。
- --seeds 0,1,2,3：以逗号分隔的随机种子列表。
- --benchmark split|permuted：基准选择。
- --methods NaiveSGD,EWC,...：要运行的方法子集。
- --device cuda|CPU：设备选择（默认自动检测）。

方法特定配置（完整模式）：

• 无上下文基线：5 个训练轮次，每个类别 500 个训练样本 / 200 个测试样本。

• CFlow：80 个回合、3 个流层、隐藏维度 256、上下文维度 32、每个任务重放 100 次。

• HyperNet：200 个回合、秩 64、隐藏维度 256、上下文维度 64、direct_stats=True、base_drop_rate=0.3、contrastive_weight=0.1。

探测协议在脚本中实现：

• **P5（错误上下文）：** 将任务 k 的上下文替换为任务（$k+1 \bmod K$）的上下文，并测量准确率下降。

• **P5b（随机上下文）：** 使用随机高斯向量作为上下文。

• **P6（随机 θ_0）：** 将 $\theta_{\text{基座}}$替换为随机初始化。

• **P7（零上下文）：** 将上下文设为零向量。

有效秩计算。对于 HyperNet 的$\mathbf{V}$ 矩阵，通过奇异值分解计算有效秩：

$$r_{\text{eff}} = \exp\left(-\sum_i \bar{s}_i \log \bar{s}_i\right), \qquad \bar{s}_i = \frac{s_i}{\sum_j s_j} \tag{85}$$

其中 $\{s_i\}$ 是$\mathbf{V}$的奇异值。这为上下文通道所使用的“有效维度数”提供了连续度量。