Jingfeng

测试时训练与适应

从分布偏移到部署中的学习与可靠评测

Jingfeng · 2026 年 10 月 22 日修订



内容摘要　本书从分布偏移与风险出发，以信息可见性、可更新状态和预测时序为统一框架，讲解测试时训练与适应的目标、算法、失败机制和可靠评测。内容覆盖归一化、熵与辅助任务、连续教师和记忆、视觉语言模型及语言模型，并以手算反例和可复现小型图像实测连接推导与实现。

> 部署中的学习，首先是信息与时序的问题。只有说清楚模型看见了什么、修改了什么，以及在何时接受评价，才有理由讨论适应是否有效。

版本说明　本书以 2026 年 9 月 3 日的 TTT/TTA 讲义为底稿，完成全文重组、术语与技术纠错、实验增补及书籍排版。文献核验截至 2026 年 10 月 22 日；该日期表示本次核验时间，不表示已穷尽此前所有研究。书中明确区分数学算例、论文机制、本次小型实测与尚未执行的扩展方案。

阅读前提　基础概率、偏导与梯度下降、分类模型和 Python。实现章节需要理解自动微分及训练/评估模式；不要求事先了解领域适应。

前言：先问是否应当学习

一个在晴天道路上训练的分类器，遇见雾天后可能变差。直觉告诉我们，应当利用新图像调整模型；但新图像没有类别标签，模型究竟凭什么知道调整方向？如果它把行人认成路标，又把这个错误当作训练目标，更多更新反而会扩大错误。测试时训练与适应讨论的正是这一矛盾：输入揭示了环境变化，却未必揭示任务的正确答案。

本书把部署模型看作带状态的系统。模型参数只是一类状态，归一化统计、教师、提示、缓存和优化器同样可能影响下一次预测。我们围绕四个问题展开：可用信息是什么，反馈为何可能有用，更新怎样控制，以及结果如何接受可信评价。由此，读者可以理解不同方法的机制，也能识别论文结果背后的额外条件。

本次修订将原题中的“更新时训练”统一改为“测试时训练”（Test-Time Training, TTT）。术语修正之外，更重要的是不再用缩写代替协议。使用同一个 Tent 更新器，先预测后更新和先更新后预测得到的是两种不同问题；使用同一个 CLIP 模型，写入缓存和反向传播提示也不是同一种成本。

怎样阅读

先读第 1 章，建立风险、信息和时序的坐标；第 2 章给出代理损失的推导与反例。随后用第 3 章的状态清单读懂最小实现，再读第 4 章的长期稳定机制。第 5 章把这些约定落实为本次真实执行的小型图像实验，保留不利结果和预算差异。

第 6、7 章将同样的问题带入视觉语言模型和语言模型；第 8 章讨论有约束的部署情境。第 9 章可以作为提交实验报告前的检查表，第 10 章及附录提供研究设计、练习与解题提示。

读者若主要关心数学，可先跟完二分类熵梯度与风险变化界；若主要关心实现，可从源模型冻结基线开始，一次只引入一个状态变化。完整的微调方法与推理服务架构分别属于《现代小模型微调》和《大语言模型原理与推理系统》的主题。本书在涉及这些内容时只解释与测试时信息和状态有关的部分，不依赖其他分册的具体页码。

证据的层次

带引用的算法描述对应文献的机制与条件；本书自行推导的等式写明变量和假设；第 5 章报告本次运行结果。对自动驾驶、医学和机器人系统的分析是实验与部署约束的设计建议，不能当作这些场景的临床或实机验证。尚未执行的扩展实验列出方案，不填入预期准确率。

在无标签环境下，“更自信”容易测量，“更正确”却需要独立证据。我们从这一差别出发，也以这一差别检验全书的结论。

目录

前言：先问是否应当学习 **ii**

第 1 章 问题、信息与时序 **1**

1.1 分布变了，风险就一定上升吗 1
1.2 不要让术语替代实验条件 1
1.3 信息集合与可更新状态 1
1.4 先预测，还是先学习 2
1.5 重置与混合分布是独立的轴 3

第 2 章 无标签反馈：推导与反例 **4**

2.1 辅助任务如何连接主任务 4
2.2 条件熵与二分类梯度 4
2.3 三个看起来相似、含义不同的熵 5
2.4 伪标签、一致性与反馈污染 6
2.5 双层训练不是联合辅助训练的别名 6

第 3 章 从归一化到可审计的更新器 **7**

3.1 BatchNorm 中有两种状态 7
3.2 训练模式不等于允许训练 7
3.3 重估统计量依赖什么假设 8
3.4 Tent：少量参数与强反馈假设 8
3.5 EATA：筛选和抗遗忘是两件事 9
3.6 SAR：可靠区域与恢复条件 10
3.7 实现前后的核验 10

第 4 章 连续适应：教师、记忆与恢复 **11**

4.1 一个更新步骤如何成为长期问题 11
4.2 CoTTA 把哪几种稳定机制组合起来 11
4.3 平滑、锚定和恢复控制不同的时间尺度 12
4.4 记忆帮助谁，又可能污染谁 12
4.5 无梯度方法仍然改变状态 12
4.6 什么时候应停止适应 13
4.7 用回访流区分适应与遗忘 13

第 5 章 最小图像实验：让协议接受检查 14

5.1 这次实验要回答的有限问题 14
5.2 数据、源训练与模型 14
5.3 固定多重集，再改变顺序 14
5.4 三种方法和两个重置设置 15
5.5 结果：统计改变比一步梯度更显著 15
5.6 代价与可追踪记录 18
5.7 独立检查和未运行的扩展 19

第 6 章 视觉语言模型：提示、低秩增量与缓存 20

6.1 先固定零样本分类的定义 20
6.2 TPT：单图像增强上的边际熵 20
6.3 提示更新与低秩更新改变了不同的对象 21
6.4 TDA 与无梯度记忆：权重冻结仍有历史 22
6.5 一个能回答问题的 VLM 实验 22

第 7 章 语言模型：学习状态与计算预算 24

7.1 先把六种机制分开 24
7.2 输入语言建模损失提供了什么反馈 24
7.3 快速权重需要训练期准备 25
7.4 验证器提供的是额外监督通道 25
7.5 计算预算应与学习收益分别报告 26

第 8 章 应用约束：何时继续、停止与恢复 27

8.1 先填写一份部署约束表 27
8.2 自动驾驶：相关帧与截止时间 27
8.3 医疗影像：患者独立性与可用概率 28
8.4 传感器流：因果时间窗与延迟反馈 28
8.5 机器人：更新改变了将来会看见什么 28
8.6 停止适应是需要验证的决策 29

第 9 章 可靠评测：顺序、恢复与累计代价 30

9.1 先冻结实验契约，再运行方法 30
9.2 把任务质量、置信度与时间分开 30
9.3 遗忘、回访与恢复不是同一个数 31
9.4 四种顺序如何构成可解释的压力测试 32
9.5 累计代价不能只写成每步耗时 32
9.6 消融应能排除一个解释 33

第 10 章 从失败模式到可检验的研究问题 **34**

10.1 先定义“不更新”的价值 34

10.2 一个可以完整执行的门控研究 34

10.3 四个有明确边界的扩展方向 35

10.4 实验包应让别人重建状态轨迹 35

附录 A 术语、符号与练习提示 **36**

A.1 术语对照 36

A.2 符号与记录单位 37

A.3 练习：从推导到协议审查 38

A.4 按已有基础安排实践 39

参考文献 **40**

第 1 章　问题、信息与时序

1.1　分布变了，风险就一定上升吗

在源域上训练分类模型时，令 $P_s(X,Y)$ 表示联合分布，f_{θ_0} 表示部署前固定的模型。给定损失 ℓ，源风险与第 t 个部署环境中的目标风险分别是

$$R_s(\theta)=\mathbb{E}_{P_s}\ell(f_\theta(X),Y),\qquad R_t(\theta)=\mathbb{E}_{P_t}\ell(f_\theta(X),Y). \tag{1.1}$$

下标 s 表示源域，下标 t 同时标记时间及当前目标分布；讨论固定目标域时可将其视作常数。$P_t\neq P_s$ 只表示分布不同，并不推出 $R_t(\theta_0)>R_s(\theta_0)$。变化可能集中在模型不使用的背景，也可能让问题变得更容易。需要测量的是固定源模型的任务风险，而非分布差异本身。

“协变量偏移”通常指 $P_s(X)\neq P_t(X)$ 而 $P_s(Y\mid X)=P_t(Y\mid X)$；这里的相等是建模假设，不应从图像看起来相似直接推断。“标签偏移”则要求 $P_s(Y)\neq P_t(Y)$ 且 $P_s(X\mid Y)=P_t(X\mid Y)$。若只知道类别比例变化，却不知道类条件分布是否保持，应该说“标签先验发生变化”，而不要自动套用标签偏移估计公式。

当 $P_t(Y\mid X)$ 变化时，旧决策边界可能不再正确。例如同一传感器读数在维护前后代表不同故障，单纯把输入统计移回源域并不能恢复语义。开放集又增加了标签空间变化：若 $\mathcal{Y}_t\not\subseteq\mathcal{Y}_s$，强制在源类别中选择答案会把未知类别错误压进已知类别。

例 1.1 (无标签适应的不可辨识性). 设目标输入总是同一幅图像 x^*。环境 A 的真实标签恒为 0，环境 B 恒为 1；两者的无标签边际 $P_t(X)$ 完全相同。只访问 x^* 的适应算法，在两种环境中接收的信息相同，因而不可能同时保证比同一个固定分类器更好。这一反例并非说适应无用，而是指出改善必须依赖某种额外结构：语义保持增强、聚类假设、训练期辅助任务、历史约束或真实反馈。

1.2　不要让术语替代实验条件

测试时适应（Test-Time Adaptation, TTA）广义地利用部署输入调整预测系统；“完全测试时适应”（fully test-time adaptation）强调在适应阶段不访问源训练数据。源数据参与过模型训练，与部署阶段仍能访问源数据，是两种不同的信息条件。源模型本身已携带训练信息。

测试时训练（Test-Time Training, TTT）的经典路线先在源训练期联合主任务和自监督辅助任务，再在测试时利用辅助损失更新共享表示 [1]。后续文献也将 TTT 用于架构中的可学习记忆或部署后自监督训练，因此写报告时必须说明是何种机制，不能仅凭标题归类。

上下文学习（in-context learning, ICL）在不更新全局参数的情况下利用上下文示例改变预测。若示例带有任务标签，它就使用了监督信息，即使没有反向传播。普通 KV cache 复用已算出的表示，既可能帮助 ICL，也可能只是减少重复计算；缓存存在并不表示系统获得了新的学习信号。第 7 章进一步区分这些情况。

1.3　信息集合与可更新状态

定义 1.1 (部署状态). 在第 t 次观测之前，令

$$s_t=(\theta_t,b_t,\bar\theta_t,\mathcal{M}_t,\phi_t,o_t,r_t), \tag{1.2}$$

表 1.1 相邻范式按信息与修改对象区分；同名方法仍须另报时序。

范式	关键可用信息	发生的变化
领域泛化 DG	训练时单源或多源数据，部署时冻结	在部署前学习泛化表示
无监督领域适应 UDA	训练/适应期访问源数据与无标签目标集	联合利用两域，可离线多轮
源数据不可访问适应 SFDA	源模型和无标签目标数据	不必严格在线；可能访问完整目标集
经典 TTT	源训练时准备辅助任务，测试时无任务标签	辅助任务梯度修改共享表示
完全 TTA	源模型、目标输入及事先约定的统计/状态	修改统计、参数、提示或缓存
连续 TTA	状态随流保留，目标环境可变化	长期积累、遗忘与恢复均须评估
带监督在线学习	当前或延迟到达的真实标签	使用明确监督，不能称为无标签 TTA
测试时计算 TTC	查询、固定模型、可能的验证器	增加采样/搜索预算，不必改变模型

分别包含模型参数、归一化缓冲区、教师参数、记忆/缓存、提示或低秩增量、优化器状态和随机数状态。某方法没有使用的分量可省略。能影响后续输出且可能随时间变化的对象都应进入状态清单。

令 B_t 为当前到达的一组输入，$\mathcal{I}_t^-$ 为它到达前可用的信息，包括源模型、允许保存的源统计、过去观测和公开配置；到达后的信息为 $\mathcal{I}_t = \mathcal{I}_t^- \cup \{B_t\}$。无标签适应器应满足

$$s_{t+1} = \mathsf{U}(s_t, B_t; \mathcal{I}_t, \xi_t), \qquad Y_t \notin \mathcal{I}_t, \tag{1.3}$$

其中 ξ_t 为增强等随机性。评估器可在输出固定后用真实标签计分，但不能把标签或由标签选择的阈值返回给 U。

这条边界也包括间接通道：若根据目标准确率选择最佳 checkpoint、域边界、学习率或停止步数，实验已经利用目标标签。保留样本 ID 用来追踪数据顺序是允许的；根据 ID 查出真实类别来平衡适应缓存，则需要披露额外信息。

1.4 先预测，还是先学习

1.4.1 三种常见的时间约定

严格的在线预测可以写为

$$p_t = \mathsf{F}(s_t, B_t), \quad \text{固定并计分 } p_t, \quad s_{t+1} = \mathsf{U}(s_t, B_t). \tag{1.4}$$

这称为先预测后更新（predict-then-update，也常称 prequential）。当前参数梯度不影响已提交的当前预测，却可影响下一批。

另一种约定先用当前输入更新，再重复前向预测：$s'_t = \mathsf{U}(s_t, B_t)$，$p_t = \mathsf{F}(s'_t, B_t)$。它允许当前输入上的无标签优化，但需要计入额外前向、反向及延迟。这是合法的测试时转导协议，不能把它与式 (1.4) 的结果混作同一比较。

第三种是离线目标集适应：可以多轮访问整个目标数据集，最后统一评估。即使不访问标签，它仍比不可回看的流式协议拥有更多信息。

1.4.2 BatchNorm 的隐藏转导性

“参数更新前预测”不等于“完全未使用当前批统计”。若前向传播读取当前 B_t 的均值与方差，样本 x_i 的预测会依赖同批其他样本。按时间先后分组时，整批预测还意味着等待批内最后一个输入。如果应用要求逐帧立即输出，则必须使用历史窗口或单帧可计算的统计，而不能把未来帧偷偷装进当前批。

因此本书把参数更新时刻与统计可见性分开记录。第 5 章的 BN 与 Tent 风格实验是“当前批统计、参数更新前输出”的批转导协议；不是严格逐样本因果在线预测。

1.5 重置与混合分布是独立的轴

Episodic 表示每个评估单元从相同初始状态开始。单元可以是单图、批或一个预定义短序列，须明确声明。重置应覆盖式 (1.2) 中的所有持久分量；只恢复权重却保留优化器动量，仍会泄漏历史。

Continual 表示适应状态在连续单元间保留；是否可回看旧样本、是否给定域边界、是否更新教师，都需另列。Mixed-domain 表示同一批或短窗口可能含不同域，它描述数据组织而不是重置策略。一个实验可以同时是 continual、混合域且先预测后更新。

关键判断

一句足够清楚的协议描述应类似：“不访问源数据或目标标签；批大小 16；当前批统计参与预测；提交更新前输出后进行一步 affine 梯度更新；模型、优化器和教师在整个流中保留；不使用域边界，不回看原始图像。”方法名称只能放在这些约定之后。

本章给出的约定还不能说明无标签反馈为何有用。下一章将直接分析损失，区分可计算的代理目标与真正关心的任务风险。

第 2 章　无标签反馈：推导与反例

2.1　辅助任务如何连接主任务

将共享编码器、主任务头和辅助头参数写为 $\theta_e, \theta_m, \theta_a$。经典 TTT 的源训练目标为

$$\min_{\theta_e,\theta_m,\theta_a} \mathbb{E}_{(x,y)\sim P_s}\big[\ell_m(f_{\theta_e,\theta_m}(x), y) + \lambda \ell_a(g_{\theta_e,\theta_a}(a(x)), r(x))\big]. \tag{2.1}$$

$a(x)$ 为输入变换，$r(x)$ 为自监督可知的目标，例如旋转角或被遮挡像素。源训练建立了共享表征，测试时才可能通过辅助头的梯度改善主任务。单纯给任意模型附加一个未训练的辅助头，不具备这一前提 [1]。

部署时固定主任务头，对辅助损失做 K 步更新。每一步的方向与风险下降之间存在一个可检查但通常无法在无标签流上直接观测的条件。

命题 2.1 (局部梯度对齐与步长). 固定样本的主任务损失 L_m 对参数具有 L-Lipschitz 梯度。令 $g_m = \nabla L_m(\theta)$、$g_a = \nabla L_a(\theta)$，一次辅助更新满足

$$L_m(\theta - \eta g_a) \le L_m(\theta) - \eta \langle g_m, g_a\rangle + \frac{L\eta^2}{2}\|g_a\|^2. \tag{2.2}$$

若 $g_a \neq 0$ 且内积为正，则取 $0 < \eta < 2\langle g_m, g_a\rangle/(L\|g_a\|^2)$ 可使右侧严格小于 $L_m(\theta)$。

证明. 由光滑函数的下降界 $L_m(\theta + d) \le L_m(\theta) + \langle g_m, d\rangle + L\|d\|^2/2$，代入 $d = -\eta g_a$ 即得。该条件针对当前点及当前损失，不是跨域总体风险保证。 □

原稿把“梯度内积平均为正”直接当作充分信号，省略了二阶项与步长。式 (2.2) 说明，方向正确仍可能因步长太大而变差；目标域上没有 y，也无法直接获得 g_m。训练期相关性、验证集上的机制检验和源锚定可以提供依据，却不能代替部署中的风险观测。掩码重建型 TTT 同样受此约束：恢复输入像素与识别目标类别只是在某些任务中相关 [2]。

2.2　条件熵与二分类梯度

对 C 类分类器，令 $p_c = p_\theta(Y = c \mid x)$，自然对数下的熵为

$$H(p) = -\sum_{c=1}^{C} p_c \log p_c. \tag{2.3}$$

其取值为 $[0, \log C]$，单位为 nat。Tent 最小化当前批预测熵的均值 [3]。一般参数梯度为 $-\sum_c (\log p_c + 1)\nabla_\theta p_c$。这在自动微分上很简单，却不包含真实类别。

以二分类正类概率 $p = \sigma(z) = 1/(1 + e^{-z})$ 为例：

$$H(p) = -p\log p - (1-p)\log(1-p), \tag{2.4}$$

$$\frac{\mathrm{d}H}{\mathrm{d}p} = \log\frac{1-p}{p} = -z, \tag{2.5}$$

$$\frac{\mathrm{d}H}{\mathrm{d}z} = -z\,p(1-p). \tag{2.6}$$

若 $z > 0$，梯度下降会增加 z；若 $z < 0$，则进一步减小 z。模型因此沿已有判断变得更自信，而不是沿真实标签移动。在 $z = 0$ 处熵最高但梯度恰好为零；在饱和区梯度也趋近零。“熵大”并不意味着当前 logit 上的梯度一定大。

例 2.1 (错误而自信的预测)**.** 真实标签取 $y = 0$，但模型给正类概率 $p = 0.9$，则 $z = \log 9 \approx 2.1972$。此时 $H \approx 0.3251$，$\mathrm{d}H/\mathrm{d}z \approx -0.1978$。取 *logit* 空间一步学习率 $\eta = 1$，更新为 $z' \approx 2.3950$，于是 $p' \approx 0.9164$、$H' \approx 0.2874$。真实标签的负对数似然却从 $-\log 0.1 \approx 2.3026$ 上升到约 2.482。分类结果始终错误，置信度更高。

这是直接把 logit 当作可更新变量的手算教学反例，不是 Tent 在神经网络上的性能实验。它已经足以反驳“降低熵必然提高准确率”；若所有样本都发生这一现象，批平均熵同样会降低。第 5 章的代码用有限差分独立检查式 (2.6)。

2.3 三个看起来相似、含义不同的熵

2.3.1 同一图像的增强边际熵

给一个图像构造 M 个语义保持增强，令 $p_j = p_\theta(\cdot \mid a_j(x))$，$\bar{p} = M^{-1} \sum_j p_j$。有精确恒等式

$$H(\bar{p}) = \frac{1}{M} \sum_j H(p_j) + \frac{1}{M} \sum_j D_{\mathrm{KL}}(p_j \| \bar{p}). \tag{2.7}$$

右侧第二项是视图间的广义 Jensen–Shannon 散度。证明只需展开 KL 并利用均值：

$$\frac{1}{M} \sum_j \sum_c p_{jc} \log \bar{p}_c = \sum_c \bar{p}_c \log \bar{p}_c.$$

以 $H(\bar{p})$ 为目标会同时惩罚平均不确定性与视图分歧；总和降低不保证两项分别降低；相较之下，仅最小化各视图熵，允许不同视图各自自信地预测不同类别。这是 MEMO 和 TPT 采用边际熵的关键理由 [4, 5]。

这里的“边际”跨的是同一输入的视图。如果裁剪掉目标、旋转改变数字含义或颜色本身决定类别，语义保持假设便被破坏。一致地犯错仍可能使两项都很小。

2.3.2 跨样本的类别多样性

另一个目标以不同输入的预测均值 $\bar{p}_B = |B|^{-1} \sum_{x \in B} p_\theta(\cdot \mid x)$ 构造

$$L_{\mathrm{IM}} = \frac{1}{|B|} \sum_{x \in B} H(p_\theta(\cdot \mid x)) - H(\bar{p}_B). \tag{2.8}$$

在经验分布上，这是负的输入 – 预测类别互信息：希望每个输入有明确判断，同时整个批覆盖不同类别。它和式 (2.7) 的边际熵符号不同，因为一个鼓励跨样本多样，另一个鼓励同样本视图一致。

若当前批确实只有一个类别，最大化跨样本边际熵会强行分散类别。对偏斜标签先验，要求接近均匀的预测比例可能损害任务。直方图是诊断信息，不能在未知真实先验时作为“类别越平衡越正确”的判据。

2.3.3 温度与校准

对 logits 使用温度 $T > 0$，$p = \mathrm{softmax}(z/T)$。减小 T 可在不改变 argmax 的情况下压低熵。准确率完全不变，概率校准和 NLL 却可能变差。因此任何以熵作为核心证据的实验，应同步报告概率指标及温度设置。

2.4 伪标签、一致性与反馈污染

令教师在弱增强上的硬伪标签为 $\tilde{y}_i = \arg\max_c p_{\text{teacher}}(c \mid a_w(x_i))$，构造

$$L_{\text{PL}} = \frac{\sum_i w_i[-\log p_\theta(\tilde{y}_i \mid a_s(x_i))]}{\max(1, \sum_i \mathbf{1}[w_i > 0])}, \quad w_i = \mathbf{1}[\max_c p_{\text{teacher}}(c \mid a_w(x_i)) > \tau]. \tag{2.9}$$

弱增强 a_w 提供教师目标，强增强 a_s 提供学生输入，是常见设计而非所有方法的统一公式。教师目标和筛选权重通常应停止梯度，避免学生通过移动目标而降低损失。若没有样本通过门控，本步跳过更新；不能对空张量求均值然后把 NaN 写进模型。

软目标常用 $D_{\text{KL}}(q\|p)$ 或交叉熵。对停止梯度的 q，二者仅差常数 $H(q)$；但若同时对 q 求梯度，它们不再等价。在 EMA 教师中，参数变化慢只是平滑，不是独立正确性证据。教师若逐渐吸收错误学生，仍会被污染。

一致性是“某些变换应保持答案不变”的假设，不是由无标签数据自动证明的事实。重建目标也可能偏重背景；对比损失中的近邻可能跨越真正类别。算法选择之前，先说明反馈所依赖的不变性，再用受控反例试图推翻它。

2.5 双层训练不是联合辅助训练的别名

若训练过程显式优化一次测试更新后的主任务损失，可写成

$$\theta'(x;\theta) = \theta - \eta \nabla_\theta L_a(x;\theta), \tag{2.10}$$

$$\min_\theta \quad \mathbb{E}_{(x,y)} L_m(x, y; \theta'(x;\theta)). \tag{2.11}$$

外层梯度依赖内层更新映射，可能包含二阶导数；这才是明确的双层或元学习形式。式 (2.1) 的多任务联合训练并不自动等于式 (2.11)。某些方法把任务拆成支持集与查询集，外层还须声明二者是否共享输入及标签。

本章说明了反馈为何可能相关、又为何不足以保证风险改善。接下来将把抽象状态落实到归一化层，讨论哪一次前向、哪一个缓冲区实际改变了模型。

第 3 章　从归一化到可审计的更新器

3.1　BatchNorm 中有两种状态

对二维卷积特征 $z \in \mathbb{R}^{N\times C\times H\times W}$，BatchNorm（BN）通常对每个通道在 N, H, W 维上求统计量。令该通道参与估计的元素集合为 $\mathcal{B}$，则

$$\hat{\mu} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} z_i, \qquad \hat{v} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}(z_i-\hat{\mu})^2, \qquad \mathrm{BN}(z_i) = \gamma\frac{z_i-\mu}{\sqrt{v+\epsilon}} + \beta. \tag{3.1}$$

这里 γ, β 是可学习的 affine 参数；μ, v 则可能是当前批统计或训练期保存的缓冲区。两者应分开记录。更改缓冲区不需要梯度，更改 affine 参数通常需要反向传播。网络还有分类头、卷积权重等大量参数，不能因为调用了 `train()` 就默认它们都可以更新。

当 $N = 1$ 时，BN2d 仍可能有 $H \times W$ 个空间元素，因而不一定发生数值错误。但相邻像素往往高度相关，有效样本量不等于元素个数；深层若 $H = W = 1$，每通道只剩一个值，训练模式 BN 往往无法按预期工作。BN1d 对单个向量的情形又不同。因此应检查实际张量形状、归一化维度和框架行为，而不是只看输入批大小。

LayerNorm 在单样本的特征维度上归一化，GroupNorm 在单样本的通道分组与空间维度上归一化。它们避免了跨样本统计依赖，却不因此免疫类别偏斜、错误伪标签或参数漂移。换用归一化类型还涉及训练前提：将一个按 BN 训练的模型直接替换成 GroupNorm，通常不能视为不改变模型的公平测试时基线 [6, 7]。

3.2　训练模式不等于允许训练

3.2.1　三类开关分别控制什么

在常见自动微分框架中，模块模式、梯度标记和梯度上下文控制不同的事情：

- `train()/eval()` 影响 BN、Dropout 等层的前向行为；它本身不决定优化器会更新什么。
- 参数的 `requires_grad` 决定是否为该参数建立梯度；优化器参数列表还必须与它一致。
- `no_grad()` 或推理上下文关闭图记录；即使处于训练模式，BN 缓冲区仍可能在前向中改变。

一个安全的实现步骤是先冻结全模型，再显式选择允许更新的归一化 affine 参数，最后打印名称、形状和总数。分类头冻结只能限制一条漂移路径，不能保证表示仍保留原类别语义。Dropout 是否开启也必须单独设置；有意使用随机集成是合法设计，无意开启则会改变基线与时延。

3.2.2　BN 缓冲区的细节会改变方法

多数实现的训练前向使用当前批方差，但写入运行缓冲区的方差估计可能采用不同的无偏校正。`momentum` 在某些框架中表示新批所占权重，在另一些语境中表示旧状态权重，必须通过公式消除歧义。关闭 `track_running_stats` 常用于让每次前向都使用当前统计；如果保留缓冲区，必须说明它是否在输出时被读取。

表 3.1 归一化基线与更新器的状态差别。具体库的 BN 默认值须查对应版本。

方法	前向读什么	前向后写什么	需要的梯度
冻结源模型	源模型参数与源统计	无持久写入	无
当前批统计	当前批 $\hat{\mu}, \hat{v}$，源 affine	可选择只临时读取；若累积缓冲须另报	无
Tent 风格	当前批统计及当前 affine	选定 affine 与优化器状态	预测熵对 affine
历史统计法	过去窗口或指数统计	均值、方差及样本数/权重	可无梯度

只对源模型调用一次 `eval()`，再对个别 BN 调用 `train()`，可以使 Dropout 保持关闭；但若模型包含自定义归一化、同步 BN 或嵌套模块，仍需检查实际前向路径。检查点不仅保存参数，也应保存缓冲区；恢复参数而遗漏统计量，不是完整恢复。

3.3 重估统计量依赖什么假设

AdaBN 以目标统计替换源归一化统计，试图消除表示中的环境差异 [8]。假设同一通道的目标特征约为 $z_t = az_s + b$，其中 $a > 0$，方差非零且目标统计估计准确；暂忽略式 (3.1) 中用于数值稳定的 ϵ，那么

$$\frac{z_t - \mu_t}{\sigma_t} = \frac{az_s + b - (a\mu_s + b)}{a\sigma_s} = \frac{z_s - \mu_s}{\sigma_s}.$$

这给出一个清楚的有利情形；它没有证明真实网络每层都满足同样的仿射变化，也没有保证目标批反映整域分布。若当前批只有一个类别，类别语义本身造成的均值可能被当作环境偏移而消去。

3.3.1 平均组内方差不是混合方差

设两个群体的权重为 $w, 1-w$，均值为 μ_1, μ_2，方差为 v_1, v_2。混合分布满足

$$\mu = w\mu_1 + (1-w)\mu_2, \qquad v = wv_1 + (1-w)v_2 + w(1-w)(\mu_1 - \mu_2)^2. \tag{3.2}$$

最后一项是群体均值差异引起的方差。原稿把均值与方差分别做指数平均，容易让读者误以为得到精确混合统计；只有均值漂移可忽略或目标就是平滑组内方差时，这种写法才符合其含义。

若确实要合并两个有限样本集合，记样本量 n_1, n_2，并保存平方离差和 $M_{2,j} = \sum_i (z_i - \mu_j)^2$，则

$$\delta = \mu_2 - \mu_1, \quad n = n_1 + n_2, \tag{3.3}$$

$$\mu = \mu_1 + \frac{n_2}{n}\delta, \qquad M_2 = M_{2,1} + M_{2,2} + \frac{n_1 n_2}{n}\delta^2. \tag{3.4}$$

最后用 M_2/n 或 $M_2/(n-1)$ 得到声明好的方差。精确汇总不是总比遗忘更好：在持续漂移中，全部历史可能已经过时。窗口长度或衰减系数定义了系统的时间尺度，应在独立验证流中选择。

3.4 Tent：少量参数与强反馈假设

Tent 对当前批平均预测熵做梯度更新，典型配置只优化归一化层 affine，并使用测试批统计 [3]：

$$L_t(\gamma, \beta) = \frac{1}{|B_t|} \sum_{x \in B_t} H(p_{\gamma,\beta}(\cdot \mid x)), \qquad (\gamma, \beta) \leftarrow (\gamma, \beta) - \eta \nabla L_t. \tag{3.5}$$

这一选择降低了更新参数与优化器存储，保留了源模型的大部分权重；代价仍包括通过网络计算损失梯度所需的前向、激活与反向。不能用“只更新很少参数”推断反向计算量也同比例减少。

算法 1 固定预测时刻的 Tent 风格更新器

输入：源状态 s_0；只含允许参数的优化器；无标签批流 B_t

for 每个 B_t **do**
  if episodic **then**
    恢复参数、缓冲区、优化器及声明的随机状态
  end if
  用约定的 BN 统计与 Dropout 模式前向，得到 p_t
  将 p_t 交给独立评估器；当前输出至此固定
  清零允许参数的梯度，计算 $L_t = \text{mean}\, H(p_t)$
  if 损失有限且门控允许 **then**
    反向传播，检查允许参数的梯度是否有限
    if 梯度有限 **then**
      优化器只更新归一化 affine
    else
      丢弃梯度并记录数值异常
    end if
  else
    跳过本步，并记录原因
  end if
end for

算法 1 特意选用更新前输出，目的是让时序可检查；这是本书实验的约定，并不宣称所有 Tent 代码采用同一读出。若要让当前图像受益于更新，需要再做一次前向并明确使用更新后结果。把返回的更新前 logits 误当作更新后结果，是很常见的实现与论文描述不一致。

Episodic 的重置单元也会影响解释。如果每批恢复、只做一步更新、输出又固定在更新前，则本批 affine 梯度不会影响任何已提交预测；下一批开始时它又被丢弃。此时 Tent 风格与只用当前批统计的预测应当完全一致。这是一个非常有用的单元检查，而不是必须“修成有提升”的异常。第 5 章实际验证了这一边界。

3.5 EATA：筛选和抗遗忘是两件事

EATA 通过可靠性与非冗余筛选减少无效更新，并加入抗遗忘约束 [9]。低熵门控可以排除不确定样本；用当前输出与历史移动平均输出的相似度判断冗余，可以减少几乎重复的信息。门控依赖模型自己的概率，因此高置信错误仍可能通过，历史平均也可能已经偏移。

用 w_i 表示依照当前预测与历史状态得到的选择/加权系数，其熵部分可以抽象为 $\sum_i w_i H(p_i)$。这只是机制记号；复现时应遵循原实现的归一化、阈值和权重计算，不能把任意低熵过滤都称为完整 EATA。没有样本通过时应跳过适应，并记录通过率；跳过是一个明确的输出路径。

抗遗忘部分常写成 Fisher 加权锚定：

$$L = L_{\text{selected}} + \lambda \sum_j F_j(\theta_j - \theta_{0,j})^2. \tag{3.6}$$

这里 F_j 表示在参考数据上估计的参数重要性。EATA 使用少量无标签的同分布（ID）参考样本，以源模型产生的硬伪标签估计 Fisher；这些参考样本不要求是原源训练样本。因而必须分别披露“适应

时不访问源训练数据”和“部署前或初始化时另有 ID 参考样本、或已有预计算的重要性文件”。若部署方只有权重、没有参考样本或预计算 Fisher，不能假称拥有同一个抗遗忘部件。将 F_j 统一设为常数得到的是不同的源锚定变体。

3.6 SAR：可靠区域与恢复条件

SAR 针对动态环境中的不可靠熵梯度，结合可靠性筛选、熵的锐度感知优化与恢复机制，并考察不同归一化配置 [10]。一个教学化的局部鲁棒目标为

$$\min_{\theta} \max_{\|\epsilon\| \le \rho} L_{\text{ent}}(\theta + \epsilon). \tag{3.7}$$

常见的一阶近似先沿损失梯度作小扰动，再在扰动点求下降方向；这需要额外的梯度计算。原论文只优化归一化层 affine，并非对全模型做锐度更新；采用 GN/LN 的模型不依赖跨样本 BN 统计，其训练前提仍须与 BN 模型分别说明。筛选、扰动半径、允许参数、归一化层以及最终恢复规则共同定义方法，不能把“加了梯度裁剪”当作 SAR 的同义词。

SAR 原论文用熵损失的移动平均作诊断：当其低于较小阈值时，将可更新参数恢复为源值，试图中止低熵塌缩；不是简单地“熵越高越触发恢复”。实现还应核对优化器及诊断移动平均如何重置，并与所用版本一致。某些低平均熵状态可能是正确的简单输入，也可能是单类错误；阈值需要独立数据和失败流验证。恢复一次也不意味着当前环境已经回到源分布。恢复后可能仍应采用冻结输出、等待可信窗口，或交给人工复核，而不是立即恢复同样的激进更新。

关键判断

Tent、EATA 与 SAR 的比较至少应对齐源检查点、可更新层、BN 统计、输出时刻与预算。然后逐项消融筛选、Fisher、锐度步骤与恢复；否则无法知道改善来自可靠反馈，还是来自更多前向、不同归一化或额外源信息。

3.7 实现前后的核验

从一个冻结批开始，先断言权重与缓冲区不变；切换当前批统计后，断言仅统计读取路径改变；启用更新后，核对只有白名单参数发生变化。保存并恢复状态，再对同一输入重复预测，应在声明的数值容差内一致。若随机增强不可逐位复现，应保存随机种子并解释允许的统计波动。

代码检查还应包括：空筛选、batch 为一、全部类别相同、相同图像重复、非有限输入、禁用梯度时的缓冲区变化，以及连续更新后恢复是否完整。它们分别验证协议边界，不应仅用“损失能够下降”作为实现正确的证据。接下来把状态保存得更久，讨论教师、记忆与恢复如何彼此影响。

第 4 章　连续适应：教师、记忆与恢复

4.1　一个更新步骤如何成为长期问题

连续流中，状态 s_{t+1} 既影响下一次预测，也影响下一次产生的伪标签；反馈的错误因此可以自我传播。短期提高置信度的方向，经过许多步可能抹去旧域知识。目标分布再次回访时，系统面对的已不再是原先的源模型。

设第 t 步状态相对源模型的变化为 $\Delta_t = \theta_t - \theta_0$。在从 θ_0 开始、仅做无动量 SGD、没有恢复或投影时，$\Delta_t = -\sum_{u<t} \eta_u g_u$；即使单步更新很小，总位移也可能持续累积。带动量、恢复或其他优化器时，应按真实更新映射记录位移，不能直接套用这个和式。梯度方向相关时，更新次数比单步范数更能暴露风险；反复出现的近重复帧又会放大这种相关性。因此应记录累计步数、实际接受的样本数、参数位移及重置次数，而不只记录学习率。

4.2　CoTTA 把哪几种稳定机制组合起来

CoTTA 使用教师预测指导学生，并以权重平均、增强平均和随机恢复降低长期误差积累 [11]。令学生参数为 θ_t，教师参数为 $\bar{\theta}_t$，教师在学生更新后按

$$\bar{\theta}_{t+1} = \alpha \bar{\theta}_t + (1-\alpha)\theta_{t+1}, \qquad 0 \le \alpha < 1 \tag{4.1}$$

进行指数移动平均。这里 α 是旧教师的权重；一些实现以 $1-\alpha$ 命名 momentum，必须写清公式。教师的目标通常停止梯度，学生最小化对这些软目标的交叉熵或一致性损失。

对当前输入的多个语义保持增强取教师预测平均，可以降低对单次增强的敏感性；CoTTA 还利用源模型置信度决定何时需要增强平均。原论文算法 1 的当前输出来自本次更新前的教师目标；学生更新后先进行教师 EMA，再随机恢复学生。教师并不随这次学生随机恢复立即回到源值。原论文还更新全部可训练参数，而不只更新 BN affine；若限制为 affine，应作为变体报告。具体门控和增强数量仍需按实现报告。这种设计用更多前向换取更稳定的目标，不能把额外计算视作免费。

随机恢复以一个按元素采样的掩码 m_t 将部分学生参数重新置为源值：

$$\theta_{t+1} \leftarrow (1-m_t) \odot \theta_{t+1} + m_t \odot \theta_0, \quad m_{t,j} \sim \mathrm{Bernoulli}(r). \tag{4.2}$$

它为持续漂移保留回源通道；保留源副本也带来存储开销。若把教师 EMA 改到恢复之后、改变恢复层或同步清除优化器状态，都会改变原来的系统。复现实验应以论文对应代码版本为准，并把自己的更改作为变体列出。

边界与反例

EMA 教师只是学生历史的平滑，通常没有独立标签来源。若学生长期系统性错判，教师最终也会吸收错误。多个增强一致同样可能只是共享了同一偏差；随机回源又可能损伤已学到的目标知识。三者提供控制漂移的手段，没有提供任务风险单调下降的保证。

4.3 平滑、锚定和恢复控制不同的时间尺度

EMA 的有效记忆长度可由几何权重理解：过去 k 步的权重与 α^k 成正比。若一秒内处理很多高度相关视频帧，同一个 α 代表的真实时间与低频医疗检查很不一样。对变速输入流，按时间而非仅按批数设定衰减可能更合理，但同样需要评测。

源锚定在损失中加入 $\lambda\|\theta-\theta_0\|^2$ 或式 (3.6) 的加权版本，约束每一步的偏移；恢复则是不连续地改变状态。二者也不等价于遗忘因子：遗忘旧样本可以使系统快速追踪变化，源锚定却始终偏向部署前知识。选择机制应先定义变化过程：缓慢漂移、突变、周期回访，还是未知开放集输入。

一个可审计的回滚事务至少保存模型参数、BN 缓冲、教师、优化器、提示和缓存版本。只回滚学生却保留污染教师，下一步又会将错误写回；只删除原始图像而留下由其计算的原型，也未必清除了影响。状态依赖关系决定了最小可恢复单元。

4.4 记忆帮助谁，又可能污染谁

4.4.1 缓存条目不是独立监督

记忆可保存原始样本、特征、伪标签、置信度、时间戳或域标记。每种选择都有代价：原始图像可重编码，却增加存储与隐私负担；特征较轻，但编码器变化后新旧坐标可能不一致；只存类别原型容易压缩掉多模态结构。

若缓存 $\mathcal{M}_t$ 被用于下一批训练，其读写顺序应明确为

$$p_t = \mathsf{F}(s_t, B_t; \mathcal{M}_t), \quad \mathcal{M}_{t+1} = \mathsf{W}(\mathcal{M}_t, B_t, p_t).$$

先写入再读取允许当前样本参与自己的目标，造成自检索和自投票路径。它可以作为明确的转导协议，但应增加禁止自身检索的消融。标签驱动的缓存均衡会额外使用目标答案；按预测类别均衡则仍可能受伪标签错误影响。

4.4.2 容量、冗余与过期

RoTTA 等方法把数据相关性与类别不平衡纳入连续适应，利用记忆和样本权重调节更新 [12]。这类方法提醒我们，均匀采样历史不一定适合突变环境，而只留最新样本又可能丢掉罕见类别。应报告容量、替换规则、时间衰减、重复样本识别和更新频率，而不只报告一个“memory size”。

一个简单诊断是记录每类缓存占用、条目年龄、相似度和被读取次数。大量相似条目占据容量时，先做去重或时间稀疏化，再比较效果；不能把更多相关帧误当作更多独立证据。类均衡也不是无条件目标：真实流短时间只含一种故障或一个器官时，强制均匀占用可能反而降低代表性。

4.5 无梯度方法仍然改变状态

“无梯度”描述计算路径，不描述信息可见性或风险水平。可以只更新归一化统计、类别原型、核图上的标签概率，或缓存读出。LAME 以当前批特征的关系结构正则化输出，不对网络参数做常规反向更新 [13]。由于一个样本的输出依赖批内其他样本，它仍应披露批转导性；构图和迭代求解也有实际代价。

参数冻结还有一个细节：若统计、原型或缓存持续写入，系统对未来输入的映射依然变化。反过来，给固定模型增加多次采样、增强集成或搜索，即使提高准确率，也未必构成持久学习。区分状态与预算，才能把无梯度方法、提示调优与冻结多视图基线放到公平的比较中。

表 4.1 连续适应机制的作用与失效方式。

机制	想控制什么	可能失效的原因	关键消融
EMA 教师	降低短期目标波动	长期错误被平滑继承	同预算冻结教师/学生自标
增强平均	减少单视图噪声	增强删掉语义证据	冻结多视图、增强强度
源锚定	限制参数远离源点	目标最优确实远离源点	权重、重要性来源
随机恢复	防止参数持续漂移	破坏已适应表示	恢复率、层与时间
记忆回放	降低近时偏差和遗忘	错标签、过期与重复	容量、去重、TTL
无梯度读出	减少优化状态/反向	图邻居或原型也会错	同预算、顺序与空缓存

4.6 什么时候应停止适应

没有标签时，系统通常只能测量代理量：预测熵、教师分歧、表示距离、接受率、梯度范数和状态位移。它们适合作为触发进一步检查的信号，不能直接估计真实错误率。高熵可能是新难例，低熵也可能是自信塌缩。

较完整的控制策略可设四种状态：冻结输出、候选适应、受限接受和回滚。候选更新先在隔离副本进行；若发生非有限数值、位移超预算或与允许不变性明显冲突，就丢弃。对于需要真实可靠性承诺的应用，仍需独立的标注审查、延迟反馈或经验证的外部约束；代理门控不能凭空创造保证。

停止规则也必须接受评估。至少报告被跳过比例、触发次数、误触发代价和恢复耗时。在无漂移的稳定流上频繁恢复，说明策略有持续成本；在错误集中流上从不触发，说明检测与失败机制脱节。若阈值依据待报告测试标签选择，必须标为 oracle 上界，不能当作可部署策略。

4.7 用回访流区分适应与遗忘

构造 $A \to B \to C \to A$，对首次和再次出现的 A 使用同一批样本与相同顺序，可以减小内容差异的干扰。记录 A 回访后首窗口、全段和冻结基线的结果；若允许读域边界并重置，应另设“不知道边界”的实验。长期适应系统不能只交一张所有域平均准确率表。

还应做两个压力测试。其一，把同一批样本按类集中，让 BN 与伪标签面对极端先验偏斜；其二，把近重复样本连成一段，测量有效更新次数与去重的作用。同一个样本多次出现是额外观测还是回看，必须声明。第 5 章使用可追踪的受控图像流，具体展示顺序、输出时刻与重置策略如何改变结论。

第 5 章　最小图像实验：让协议接受检查

5.1　这次实验要回答的有限问题

本章报告 2026 年 10 月 22 日实际执行的受控实验。目的有三项：检查统计重估与 affine 梯度的贡献是否被混淆；验证输出时刻与重置策略的边界；观察同一多重集在不同顺序下的结果。实验使用小型灰度图像分类器，不是 Tent 原论文的标准骨干与基准复现，也不作为自动驾驶、医学或大模型有效性的证据。

冻结模型是必要的参照。若重估统计已经损害分类，仅看到 Tent 风格比 BN 高一点，不能说它改善了部署模型；若 ECE 下降而准确率大幅下降，也不能独立宣称概率更可靠。保留不利结果，正是这个最小实验的教学价值。

5.2　数据、源训练与模型

输入来自 scikit-learn 随包提供的 digits 数据：1797 张 8×8 手写数字图像，像素为 0 到 16 的整数，类别为 0 到 9。它是 UCI 手写数字数据的一个子集，创建者为 E. Alpaydin 与 C. Kaynak，数据遵循 CC BY 4.0；数据说明与许可见官方来源 [1]。脚本直接用 NumPy 读取已安装包的 `digits.csv.gz`，没有联网下载，也没有调用 scikit-learn 的训练器。

将像素除以 16。按类别分层，以固定随机种子 42 在各类中划分约 60%/20%/20%，得到源训练 1074 张、独立验证 355 张、目标基础图像 368 张。验证集仅在训练后报告，不用于学习率、步数或模型选择。划分 ID 已保存。这里是图像级随机划分，不是书写者级划分，因此不能把结果解释成对新书写者的泛化评估。

分类器为 $64 \to 48 \to 10$ 的全连接网络，在 48 维隐层加入 BN affine，再经 ReLU 与分类头。它接收展开的图像像素，仍是图像分类模型，但有意避开大型卷积网络的资源需求。源训练优化全部参数，交叉熵、SGD、学习率 0.05、最多 64 张一批、40 轮；三个初始化种子为 13、29、47。训练后用最终第一层权重在源训练集上重新计算一次均值和总体方差，作为固定源统计。该步骤不读取验证或目标样本。

三个源模型在干净目标基础图像上的准确率分别为 95.38%、96.20%、96.20%。这些数值用于描述起点，未按目标准确率选择一个种子。所有方法使用同一个种子对应的检查点，完整保存三个模型。

5.3　固定多重集，再改变顺序

令 $x \in [0,1]^{64}$，构造三个域：

$$A(x) = x, \qquad B(x) = \mathrm{clip}(0.45x + 0.20, 0, 1), \tag{5.1}$$

$$C(x) = \mathrm{clip}(x + \varepsilon, 0, 1), \qquad \varepsilon_j \sim \mathcal{N}(0, 0.25^2). \tag{5.2}$$

B 是对比度压缩并抬高背景，C 是固定实现的独立像素高斯噪声。它们是受控教学扰动，不是 ImageNet-C/CIFAR-C 的标准损坏实现。扰动种子为 20261022，三个训练种子共享同样的目标输入。

[1] https://archive.ics.uci.edu/dataset/80/optical+recognition+of+handwritten+digits；https://creativecommons.org/licenses/by/4.0/。子集说明：https://scikit-learn.org/stable/datasets/toy_dataset.html#optical-recognition-of-handwritten-digits-dataset

为检查回访，保存 A 的两个完全相同的出现版本 A_0, A_1，以及 B、C 各一份，共 $4 \times 368 = 1472$ 次观测。重复出现是显式允许的额外观测；适应器没有获得这些图像属于同一基础样本的标记。四种顺序只改变同一多重集的排列：

1. **预定交错基准序（结果文件名 original）**。对每个基础图像依次排列 $A_0(x_i), B(x_i), C(x_i), A_1(x_i)$，再处理下一图像。这里的“原序”指构造后预先约定的序列，不是 digits 原文件顺序，也不模拟自然传感器时间流。
2. **随机打乱**。对全部 1472 次观测作固定种子的随机排列，不按方法重新采样。
3. **类别集中**。按真实类别排序，类内按固定原始索引排列。这是离线评估器使用标签构造的压力测试；标签不进入适应器，不能把这种排列当作无标签部署时可获得的能力。
4. **域回访**。连续排列完整的 $A_0 \to B \to C \to A_1$。每段 368 张，恰为 23 个批；域边界不传给更新器，不触发重置。

交错基准序使一批内包含不同域和近重复输入；类别集中使当前批的类别分布极不均衡。它们故意检验 BN 的代表性假设。改变顺序还改变分批组成，因此本实验中的顺序效应包含“当前批统计不同”和“持久参数历史不同”两部分。

5.4 三种方法和两个重置设置

每批大小为 16，BN 使用每个隐层特征的当前批均值与总体方差，$\epsilon = 10^{-5}$。三种方法为：

- **Frozen**：参数与源统计都固定，不更新任何状态。
- **BN**：固定全部参数，只在当前前向中替换为当前批统计；不累计运行缓冲区。
- **Tent 风格**：同样使用当前批统计，只对 BN 的 48 维 γ, β 做一步平均熵梯度更新，SGD 学习率 0.01、无动量，不改分类头与第一层权重。

这套手写 NumPy 实现用于隔离机制，与框架中的完整 Tent 模型配置不同。学习率、训练轮数与扰动强度固定执行，没有在待报告目标标签上搜索最佳参数。

本实验固定输出当前前向、再更新 *affine*。BN 当前批统计已参与预测，因此它是批转导协议，不能称为逐样本立即输出的严格因果在线实验。Tent 风格从该前向计算熵梯度，不再做更新后第二次预测。输入读取、计分和更新分离，适应器的接口只接受图像，没有标签参数。

Episodic 在每批前恢复 affine；continual 在整个 1472 次观测中保留 affine。无优化器动量、教师和缓存，BN 统计也不持久写入，因此本实验的重置状态较少。Frozen 与 BN 的两种设置理论上应相同；episodic Tent 的更新会在下一批前被丢弃，其输出也应与 BN 相同。脚本对全部种子和顺序逐元素检查了这些等价关系。它们体现协议约束，不代表一般 episodic TTA 都不能改变当前预测。

5.5 结果：统计改变比一步梯度更显著

表 5.1 报告三个种子的均值与样本标准差。准确率、最差域准确率和 ECE 用百分数表示；最差域为 A、B、C 各域准确率的最小值，A 的两个相同版本都参与计分。ECE 采用 15 个等宽置信度区间，空区间贡献为零。NLL 使用自然对数；完整结果还保存 Brier、平均熵和最差非重叠完整 64 观测块准确率。后一个指标只检查从序列起点划分的完整块，不是所有长度 64 滑动窗口的最小值。

冻结模型在四种排列中的累计终点完全相同，是无状态基线应有的检查。BN 则在类别集中时严重退化，因为归一化把类别特有的隐层均值也消去了。交错基准序里还混合了域与重复图像，当前

表 5.1 本次连续设置的眞实结果，3 个种子；准确率、最差域与 ECE 为均值 ± 样本标准差，NLL 为均值。

顺序	方法	准确率 (%)	最差域 (%)	ECE(%)	NLL
交错基准	Frozen	90.65 ± 1.01	84.60 ± 2.34	8.82 ± 0.93	0.370
	BN	67.37 ± 1.12	37.14 ± 1.81	4.73 ± 1.82	1.040
	Tent 风格	67.57 ± 1.26	37.23 ± 2.05	4.75 ± 2.11	1.043
随机打乱	Frozen	90.65 ± 1.01	84.60 ± 2.34	8.82 ± 0.93	0.370
	BN	86.39 ± 1.09	77.63 ± 1.28	5.34 ± 0.82	0.443
	Tent 风格	86.37 ± 1.11	77.63 ± 1.28	5.17 ± 0.81	0.442
类别集中	Frozen	90.65 ± 1.01	84.60 ± 2.34	8.82 ± 0.93	0.370
	BN	13.25 ± 0.07	7.52 ± 0.42	45.38 ± 0.78	3.925
	Tent 风格	13.29 ± 0.08	7.52 ± 0.42	45.69 ± 0.83	3.949
域回访	Frozen	90.65 ± 1.01	84.60 ± 2.34	8.82 ± 0.93	0.370
	BN	89.76 ± 0.94	80.80 ± 2.99	4.14 ± 0.55	0.331
	Tent 风格	89.70 ± 0.85	80.80 ± 3.13	3.96 ± 0.54	0.330

表 5.2 每批重置、更新前输出的实测边界。

顺序	BN 与 Tent 风格准确率 (%)	ECE(%)	NLL
交错基准	67.37 ± 1.12	4.73 ± 1.82	1.040
随机打乱	86.39 ± 1.09	5.34 ± 0.82	0.443
类别集中	13.25 ± 0.07	45.38 ± 0.78	3.925
域回访	89.76 ± 0.94	4.14 ± 0.55	0.331

批统计明显损害分类。不能把这两种压力测试归纳为所有目标域统计都无用；它们表明统计重估所需的代表性假设在此不成立。

Tent 风格相对 BN 的变化很小，且没有在任何顺序的总体准确率上超过 Frozen。类别集中时，平均熵从 BN 的约 1.190 降至约 1.179 nat，NLL 却从 3.925 升至 3.949。在本次类别集中设置中，代理目标改善伴随着眞实概率损失恶化。交错基准的 BN 虽有更低 ECE，准确率和 NLL 都更差，也说明 ECE 不能单独承担可靠性结论。

5.5.1 Episodic 结果不是缺失的对照

图 5.1 保留了累计表现随顺序的变化，终点均值对应表 5.1。表 5.2 单列 episodic 的实际结果。每个格子的 BN 与 Tent 风格逐样本概率完全相同，因而准确率、ECE、NLL 等均相同；原始结果仍保留两种方法各自的独立运行记录。Frozen 仍为 90.65 ± 1.01%，ECE 为 8.82 ± 0.93%。

若把当前更新后的再前向作为输出，这一相等关系将不再必然成立，同时多出一次前向和当前输入优化时间。那是另一个合理实验。本书未运行这一扩展；主实验沿用预先固定的更新前输出时序。

5.5.2 回访后的保持与恢复

在域回访顺序中，A_0, A_1 使用完全相同的图像与内部次序，因而可以作配对比较。Frozen 首次与回访 A 均为 95.92 ± 0.47%；BN 均为 92.75 ± 0.87%。连续 Tent 风格由首次 92.75 ± 0.87% 变为回访

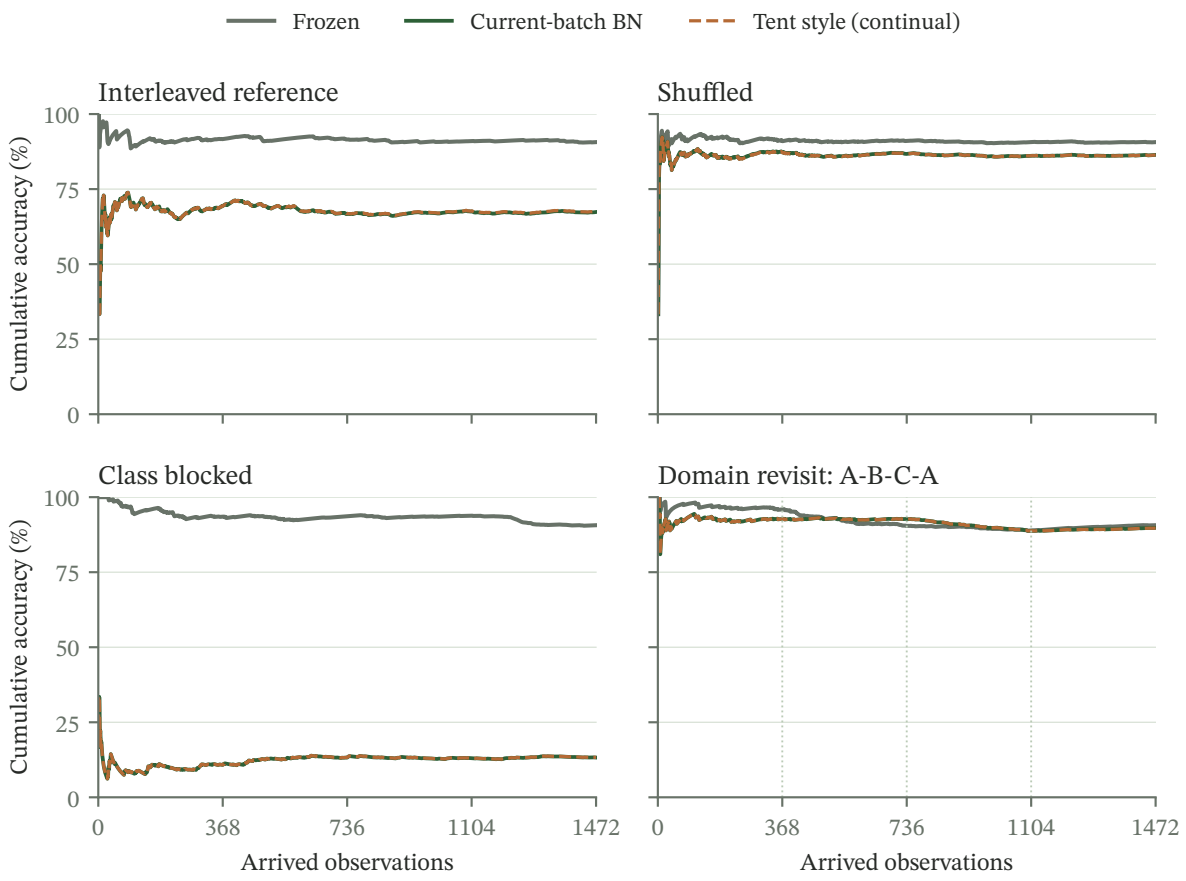


图 5.1 小型受控图像实验的累计准确率：三种子的逐位置均值，未绘制置信区间。横轴为已到达观测数，四图使用同一多重集；域回访图中竖虚线为评价器已知的段界，不提供给适应器。绿色与棕色曲线接近，表示本次一步 affine 更新相对当前批统计的变化很小。数据、绘图源码和校验和均保留；图中没有论文报告数值。

表 5.3 实验目录中的证据与用途。除源码行外，文件均位于共同结果目录。

位置	内容
`experiments/minimal_image_tta.py`	数据构造、源训练、标签隔离更新器、评估与断言
`config.json`	超参数、顺序定义、信息条件与真实命令
`environment.json`	版本、平台、数据及脚本校验和
`checks.json`	熵梯度、有限差分与高置信错误反例
`stream.npz`	固定图像、评估标签、基础 ID、划分与四种排列
`results.csv`	72 组运行的主要指标与耗时
`results.json`	主要指标、逐域结果和配对回访指标
`summary.json`	不挑种子的连续设置汇总

92.48 ± 0.42%，配对差为 −0.27 ± 0.47 个百分点；其中两个种子变化为零，一个下降 0.82 个百分点。差异很小，不足以支持显著性的论断。

回访最初 64 张的准确率，Frozen 为 97.40 ± 0.90%，BN 为 92.19 ± 1.56%，连续 Tent 风格为 91.67 ± 1.80%。这既报告了回访全段，也保留了恢复初期窗口。没有使用回访域标签触发恢复或挑选重置时刻；实验也没有设置风险阈值驱动的回滚，因此不能声称某方法“已恢复到安全水平”。若要测恢复时间，须事先定义参考准确率、窗口、容差和不恢复时的删失规则。

5.6 代价与可追踪记录

运行环境为 Apple M5、32 GiB 内存、CPU NumPy，Python 3.12.7、NumPy 2.2.6；数据包版本 scikit-learn 1.5.1，设置 `OPENBLAS_NUM_THREADS=1`。平台与可执行路径、脚本及数据文件 SHA-256 均保存在环境文件中。全部 $3 \times 4 \times 3 \times 2 = 72$ 组运行有独立结果，没有以最佳种子替代平均。

每组处理 92 个批。Frozen 与 BN 各 92 次前向、零次反向；Tent 风格 92 次前向及 92 次手写 affine 反向。以域回访连续设置为例，各种子的批内操作中位耗时再取均值，约为 Frozen 0.016 ms、BN 0.025 ms、Tent 风格 0.038 ms。计时包含该批前向及规定更新，不包含数据读取、源训练、设备传输、评估计分或持久化；没有隔离系统负载，也不是完整服务的尾延迟基准。结果文件保存逐组 p50、p95 与累计操作耗时，不能把这些微型 CPU 数字外推到大型视觉模型。

在主题的 `experiments/` 目录中，实际命令为：

```
OPENBLAS_NUM_THREADS=1 /opt/anaconda3/bin/python \
  minimal_image_tta.py \
  --output results/minimal-image-20261022 \
  --seeds 13 29 47 --epochs 40
```

换机运行应使用可用的 Python，并安装 NumPy 与提供数据文件的 scikit-learn；模型训练和适应计算只依赖 NumPy。源码不调用 scikit-learn 的二进制训练组件，以避开本机旧 SciPy 与 NumPy 2 的 ABI 冲突。依赖导入失败的复核日志另存，成功运行日志与结果分开标识。

共同结果目录为 `experiments/results/minimal-image-20261022/`；表 5.3 将重复前缀省去，完整机器记录仍保留原路径。

5.7 独立检查和未运行的扩展

有限差分用中心差分步长 10^{-5} 检查二分类熵导数；同时在独立随机小矩阵上检查 softmax 熵对 affine 的解析梯度，误差阈值分别为 10^{-9} 和 10^{-8}。脚本还检查高置信错误更新后熵下降但真实 NLL 上升，以及 Frozen 的顺序不变性、BN 两种重置的等价和 episodic Tent 与 BN 的等价。它们检验梯度与协议，不构成真实任务中适应必然有益的测试。

下一步可以保持同一数据划分，加入更新后输出、历史统计窗口、批大小一、归一化维度替换、熵筛选、源锚定和恢复机制。一次只改一个机制，并记录额外前向、持久状态与选参数据。若扩展到 CIFAR-C 或实际传感器流，应重新冻结模型、预处理与损坏版本，建立独立源验证流，加入多个流种子和任务相关概率指标。CLIP/TPT、真实医疗、机器人和大模型实验均未在本次执行；本章提供的是可以核验的起点，不是这些扩展的性能预测。

本章最直接的结论是：在相同样本上，批组成和统计可见性足以造成很大差异，而一次小 affine 更新可能只带来很小变化。先检查协议与冻结基线，再解释算法名称，才能知道究竟是哪一种信息改变了结果。

第 6 章　视觉语言模型：提示、低秩增量与缓存

视觉语言模型为测试时适应提供了一个有吸引力的起点：图像与类别名称已经位于同一表示空间，部署方可以改变类别集合和提示，而不必重新训练分类头。但“零样本”只表示没有使用当前任务的标注样本，并不表示没有可变状态，也不表示任何测试时更新都满足相同的信息约束。本章以 CLIP 为例，逐一拆开提示更新、低秩参数更新与无梯度缓存，讨论它们分别从哪里获得反馈，以及怎样验证这种反馈没有变成自我强化的错误。

6.1 先固定零样本分类的定义

设图像编码器为 g_v，文本编码器为 g_t，第 c 类的名称为 n_c，提示参数为 p。将文本序列记为 $[p; n_c]$，归一化表示为

$$v(x) = \frac{g_v(x)}{\|g_v(x)\|_2}, \qquad t_c(p) = \frac{g_t([p; n_c])}{\|g_t([p; n_c])\|_2}. \tag{6.1}$$

CLIP 的分类概率可以写成

$$z_c(x; p) = \frac{v(x)^\top t_c(p)}{T}, \qquad p_p(c \mid x) = \mathrm{softmax}(z(x; p))_c, \tag{6.2}$$

其中 $T > 0$ 是温度；实现中常保存对数 logit scale，并用其指数 $\alpha = 1/T$ 乘余弦相似度。原稿用 τ 乘内积而称其为温度，容易混淆温度与逆温度。式 (6.2) 固定使用 T 表示温度 [14]。

归一化并非可有可无。未归一化的点积同时反映方向和向量范数，改变提示后，模型可能仅通过增大范数而提高置信度。比较提示适应与零样本基线时，必须保持归一化规则、类别名称、logit scale 与图像预处理一致。若使用多模板集成，也要声明是先平均概率，还是先平均并重新归一化文本特征；两者一般不等价。

温度同样影响适应。降低 T 通常使概率更尖锐；若只调整一个对所有类别共享的正温度，类别排序不变，top-1 预测也不变，但熵、负对数似然和校准误差都可能显著变化。因此，“熵下降”可能完全来自尺度变化。一个基本消融是冻结表示，只改变温度，再与改变提示或视觉表示的结果比较。温度应在独立验证数据上确定，不能利用待报告测试集的正确标签选择。

类别名称属于任务定义，通常允许在零样本分类中使用；每张图像的正确类别、由它选出的最佳模板、人工纠正后的缓存条目则属于额外监督。若测试集类别由外部目录事先提供，应照实说明。若类别集合本身需要在线发现，就已经进入开放词汇发现或开放集识别，不能沿用封闭类别分类的全部假设。

6.2 TPT：单图像增强上的边际熵

6.2.1 从视图筛选到提示梯度

TPT 为一个待分类图像构造多个增强视图，以当前模型筛选相对可信的视图，再更新连续提示 [5]。设 $a_1(x), \dots, a_M(x)$ 是这些视图，

$$p_j = p_p(\cdot \mid a_j(x)), \qquad h_j = H(p_j).$$

一种与原方法思想一致的表述是：按照 h_j 从小到大排序，保留预先规定比例的索引集合 $\mathcal{J}$，优化

$$\bar{p}_{\mathcal{J}} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} p_j, \qquad \mathcal{L}_{\mathrm{TPT}}(p;x) = H(\bar{p}_{\mathcal{J}}). \tag{6.3}$$

筛选比例、视图数和步数都是超参数。若采用绝对熵阈值，还必须定义没有视图通过时的行为；固定保留数量可以避免空集合，却可能在所有视图都不可靠时仍强制更新。在一次反向传播中，视图索引通常视为离散选择，不对排序求导；多步更新时是否重新筛选，也应写入协议。

边际熵的作用可由下面的恒等式看清：

$$H(\bar{p}_{\mathcal{J}}) = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} H(p_j) + \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} D_{\mathrm{KL}}(p_j \| \bar{p}_{\mathcal{J}}). \tag{6.4}$$

式 (6.3) 同时惩罚平均条件熵与视图间分歧；一次更新使总和下降，并不保证每一项都单独下降。若每个视图都高度自信，却分别指向不同类别，平均条件熵很低而边际熵仍然很高。这解释了它为何不同于独立压低每个视图的熵，也说明“增强一致性”是一个需要检验的语义假设。

6.2.2 增强保留了什么，又删掉了什么

设原图是一只位于草地上的鸟。随机裁剪可能保留鸟，也可能只保留草地；后者对“草地”概念很自信，却不再包含原分类任务的充分证据。低熵筛选无法自动识别这种情况。医学影像中裁掉微小病灶、细粒度识别中遮掉关键纹理，也会产生同类问题。

增强设计应从任务语义出发。可以在源验证数据上审查裁剪覆盖率、颜色变换强度和预测稳定性，记录每种增强对目标对象的可见性影响；不能借助目标测试标签逐样本挑选“有效增强”。在目标流上只能观察视图分歧、表示距离和筛选比例等诊断量。它们能提示异常，却不是正确性的证明。

单样本提示调优通常采取 episodic 协议：从同一个初始提示 p_0 出发，利用当前图像的无标签视图更新，随后在声明好的视图上输出预测，最后恢复 p_0。若提示跨图像保留，则成为具有顺序效应的连续适应变体。两种设置应分别命名和报告，不能将连续流带来的历史信息收益记为独立单图像方法的收益。预测原图、平均全部视图、平均筛选视图，也构成不同的读出策略。

6.3 提示更新与低秩更新改变了不同的对象

连续提示是输入文本编码器的一小组可学习向量。冻结 CLIP 权重并不意味着没有反向传播：损失仍需通过文本编码器传到提示，计算图、激活和优化器状态都可能占用资源。若图像分支固定，其增强图像特征可预先计算并复用；提示更新后，类别文本特征通常需要重新计算。

低秩适应则给某些线性映射引入

$$W' = W_0 + \Delta W, \qquad \Delta W = \frac{\alpha_r}{r} BA, \tag{6.5}$$

其中 $A \in \mathbb{R}^{r\times d_{\mathrm{in}}}$、$B \in \mathbb{R}^{d_{\mathrm{out}}\times r}$，$r$ 是秩，α_r 是缩放系数。测试时冻结 W_0，仅更新 A, B。这增加了可调整的表示方向，但不会自动改善无标签反馈的可靠性。若两个因子都初始化为零，初始乘积虽为零，两个因子的梯度也都会为零；常见初始化让一个因子随机、另一个为零，使初始模型保持不变且有非零梯度路径。

TTL 是将低秩更新用于 VLM 测试时适应的具体工作，使用置信度相关目标和增强视图反馈 [15]。它与 TPT 的差别应表述为更新对象和目标设计的差别，而不能只用“参数更少”概括。更新视觉注意力映射会改变图像表示；更新文本提示会改变类别文本几何。若视觉编码器发生变化，历史缓存中的特征来自旧编码器，便还要处理新旧表示不一致的问题。

表 6.1 VLM 适应中应区分的三类更新。计算量取决于具体实现。

对象	写入方式与反馈	需要记录的额外状态与代价
连续提示	经文本分支反向传播，常用增强边际熵	初始提示、优化器、文本重编码、每图是否重置
低秩增量	在选定映射中更新 A, B，反馈可为熵或自监督目标	插入位置、秩、激活、优化器；视觉特征缓存是否失效
缓存或原型	根据嵌入、伪标签与筛选规则写入，无需网络反向传播	容量、替换规则、条目年龄、伪标签、跨样本持久性

PromptAlign 进一步引入表示统计对齐，其所需的代理源统计和初始化资源需要单独声明 [16]。这提醒我们：都被称为“测试时提示方法”的系统，可能拥有不同的源信息。公平比较不能只对齐测试阶段的标签可见性，还应列出源代理数据、预训练提示、额外模型和统计文件。C-TPT 将校准问题纳入提示适应，反映了准确率与概率可信度是两条不同的评测轴 [17]。

6.4 TDA 与无梯度记忆：权重冻结仍有历史

TDA 的全称是 training-free dynamic adapter。它使用动态键值缓存，将图像特征作为键、伪标签信息作为值，并结合正、负伪标签调整预测；其核心机制不需要测试时反向传播 [18]。原稿将 TDA 引作“LoRA-TTT”依据，混淆了缓存与低秩参数更新，新版将二者分开。

为了理解正缓存的作用，考虑一个简化读出。缓存 $\mathcal{M} = \{(k_i, u_i)\}_{i=1}^{N}$ 保存归一化视觉特征 k_i 与类别向量 u_i。令

$$a_i(v) = \exp\{-\beta(1 - v^\top k_i)\}, \qquad z_{\text{mem}}(v) = \lambda \sum_{i=1}^{N} a_i(v) u_i. \tag{6.6}$$

最终 logit 可以由零样本 logit 与 z_{mem} 组合。式 (6.6) 只展示相似性加权投票这一教学抽象，并非 TDA 包括负缓存在内的完整算法。负伪标签表达的是“此样本可能不属于某些类别”，不同于直接宣称它属于当前最高概率类别；但若概率分布本身失准，负标签同样可能错误。

没有梯度不等于没有适应风险。若一个高置信错误样本写入缓存，它可能给邻近样本提供错误证据；后续条目再沿用这些预测，污染便会扩散。低熵条目不一定正确，类别平衡缓存也不一定符合真实类别先验。重复视频帧可能大量占用容量，使缓存看似稳定、实际只记住一小段场景。删除、过期和回滚规则决定了错误能否被清除。

读写顺序还会改变当前样本的结果。先预测、后写入时，当前结果只使用过去记忆；先将当前样本写入再查询时，样本可检索到自身，等价于增加一条自投票路径。后一种做法可以是明确允许的 transductive 协议，但必须与前一种分开报告。为检验是否只在强化原预测，可以加入禁止自身检索、仅使用过去条目、打乱流顺序三个消融。

BaFTA 使用在线聚类估计类别中心，并聚合原始类别表示与适应后表示的预测，也属于无需反向传播的路线 [19]。其原型仍然是持续写入的状态，不能与每图重置的 TPT 直接视作同一时序协议。方法分类应依据实际可变对象，而非论文名称中的“training-free”或“adapter”字样。

6.5 一个能回答问题的 VLM 实验

可以从同一 CLIP 检查点、同一类别描述和同一图像预处理开始，比较冻结单视图、冻结多视图集成、episodic TPT，以及连续缓存方法。冻结多视图基线尤其必要，否则增加图像编码次数本身的

收益会被误记为更新收益。进一步比较低秩方法时，应固定可用资源，再分别报告相同视图数和相同实际时延下的结果。

评估程序可读取标签计算指标，适应程序则只接收图像、允许的类别列表与过去状态。把标签保存在独立评估器中，能够避免缓存筛选、停止步数或最佳检查点选择无意间使用目标答案。对预先构造的每一种顺序，所有方法必须看到同一批图像；不要根据某方法的表现重新排列样本。

记录 top-1、最差域、负对数似然、Brier 分数和校准曲线，还应记录提示或增量的变化范数、视图筛选率、缓存命中率、条目年龄与类别组成。预测类别分布收缩是值得检查的信号，但当真实流本就只含少数类别时，它也可能正确。最终解释需要独立标签评估，不能把类别均衡本身当作学习目标已经成功的证据。

本章没有执行 CLIP、TPT 或 TTL 的新性能实验，也不把原论文中依赖特定骨干、数据集和预算的提升数字转换为本书实测。前述设计用于明确可复现的对照；方法的收益仍需要在所选任务上验证。

练习与提示。 构造两个二分类视图 $p_1 = (0.99, 0.01)$、$p_2 = (0.01, 0.99)$，计算平均条件熵与边际熵，并解释为什么单独压低两个条件熵不能消除分歧。再构造一个含高置信错误条目的两项缓存，比较禁止与允许自身检索的结果。最后说明：若适应只改变共享温度，为什么 top-1 可以完全不变而熵显著下降？

第 7 章　语言模型：学习状态与计算预算

语言模型让“测试时发生了变化”这句话变得尤其含糊：多写一段推理、检索一条笔记、复用 KV cache、更新一个低秩矩阵，都可能改变答案，却使用不同信息、需要不同训练准备。本章沿用第 1 章的状态与信息约定，把这些变化分开讨论。重点是判断一个实验究竟验证了什么，而非把所有改进都归入 TTT。

7.1 先把六种机制分开

令 q 是当前查询，c 是允许读取的上下文。基础模型参数记为 θ_0，低秩增量为 ϕ，架构内快速权重为 W，外部记忆为 $\mathcal{M}$。这些对象都能进入广义状态 s，但“属于状态”不意味着它们具有相同学习含义。

表 7.1　语言模型部署中的六种变化。每种机制都还需声明是否跨查询保留状态。

机制	变化的对象与信息	训练前提及必须核查的边界
上下文学习	给定示例或指令改变条件分布；全局参数固定	示例标签就是监督信息；上下文长度与检索来源须计入
普通 KV cache	保存已处理 token 的键值表示	通常是同一计算的复用；不含参数梯度，也不自动形成跨任务学习
外部记忆更新	将摘要、解题策略或工具结果写入 $\mathcal{M}$	可用固定黑盒模型；记忆来源、筛选、读写成本与污染须披露
部署后参数更新	用当前文本或伪标签更新 ϕ 或 θ	需要可求导模型及优化器；普通预训练初始化也可尝试，效果须验证
架构内快速权重	内层规则把上下文编码进临时 W	通常在训练期就使用相同更新结构，学习写入、读取或适应初始化
搜索与采样	增加候选数、推理长度或验证次数	参数可完全固定；验证器有独立训练与信息来源

普通自回归解码即使使用 KV cache，仍可计算与无缓存实现相同的条件概率。它主要节省重复前向，不应仅凭“存在记忆”就报告获得了测试时学习。若系统决定保留哪些历史结论、压缩成什么规则，并让后续查询使用它们，则外部状态确实承担了经验积累；但这仍不同于参数训练。Dynamic Cheatsheet 是以可读文本维护策略和经验的例子 [20]。其核查对象包括记忆是否包含测试答案、是否跨任务共享、删除后能否恢复基础行为，以及提示和工具预算是否与基线一致。

7.2 输入语言建模损失提供了什么反馈

给定允许使用的输入序列 $c_{1:L}$ 和固定掩码 $m_j \in \{0,1\}$，令 $M = \sum_j m_j > 0$，定义

$$L_{\text{in}}(\phi; c) = -\frac{1}{M} \sum_{j=1}^{L} m_j \log p_{\theta_0,\phi}(c_j \mid c_{<j}), \qquad \text{PPL}(c) = \exp L_{\text{in}}(\phi; c). \tag{7.1}$$

掩码明确哪些位置被训练：可以只含用户提供的领域文本，排除特殊 token 和不属于训练材料的片段。TLM 研究了以输入困惑度最小化为信号、以 LoRA 为更新对象的测试时学习 [21]。这利用输入内部的自监督信号，没有获得该查询的真实回答。

对固定输入，最小化 PPL 与最小化 L_{in} 的极小点相同；直接对 PPL 求梯度会额外乘上 $\exp L_{\text{in}}$，因此优化尺度并不相同。实现通常优化平均负对数似然，而不是先指数化。比较不同长度文本时，还须说明采用 token 加权平均还是先逐文本平均。

高困惑度只能说明输入在当前模型下难以预测。罕见术语、拼写错误、乱码、对抗性文本和不熟悉的正确知识都可能产生高 PPL，不能据此断言“信息更多”或“更值得更新”。同样，模型更会复述输入，也不推出更会回答问题。可检验的假设是：某种领域文本上的自监督更新，是否改善独立查询上的任务损失，同时保持无关任务表现；这需要与本次更新文本分离的评价查询。

边界与反例

如果先用整个 $c_{1:L}$ 更新，再在同一文本上计算式 (7.1)，得到的是适应后的拟合损失。它允许使用后面的文本改变前面位置的概率，不能当作严格因果的在线困惑度。因果评价应在看见 c_j 前固定其概率，再以该 token 更新后续状态。若查询含参考答案、评分器反馈或验证集标签，必须排除这些内容或明确改称有监督适应。

7.3 快速权重需要训练期准备

一个简化的快速权重模块先从上下文 token 得到 (k_j, v_j, q_j)，以小网络 f_W 的权重承载临时记忆。以下采用先读后写约定：

$$o_j = f_{W_{j-1}}(q_j), \qquad W_j = W_{j-1} - \eta_j \nabla_W \ell_{\text{mem}}(f_{W_{j-1}}(k_j), v_j). \tag{7.2}$$

这里 W_j 是序列内状态，投影矩阵和其他慢速参数属于部署时通常固定的 θ。q_j, k_j, v_j 能使用哪些 token，要与预测目标一起声明：若目标是预测第 $j+1$ 个 token，读取已到达的第 j 个 token 是因果的；若目标是评价第 j 个 token 的概率，就不能提前使用其内容。采用先写后读或分块更新的模型也可以成立，但必须配套解释可见性掩码和训练目标。

这不是在普通冻结 Transformer 中任意插入一次梯度下降。训练期通常已经包含内层更新过程，使慢速参数学会构造可写入的表示、有效的读取方式和合适的初始化；某些设计还对更新轨迹求外层梯度。LaCT 研究分块写入与快速权重容量 [22]，说明更新粒度还影响并行性、因果结构与硬件实现。TTT-E2E 则以滑动窗口 Transformer 为基础，把长上下文的下一 token 学习与训练期元学习相结合 [23]。两者不能仅凭都使用梯度就看成同一架构。

在状态宽度、更新步数和局部计算固定时，逐块处理长度 L 的上下文可以具有随 L 线性增长的某些计算项；这不等于所有模型都获得线性总成本，更不等于任意长度信息都能无损压入固定状态。还应计算局部注意力、反向传播、优化器、分块大小及回答查询的成本，并实际测量时延和记忆容量。比较速度与利用率时，应固定任务、硬件和测量口径。

7.4 验证器提供的是额外监督通道

设固定模型为查询 q_i 生成候选 $r_{i1}, \dots, r_{iN}$，验证器给出 $v(q_i, r)$。一种选择规则是

$$r_i^* = \operatorname*{arg\,max}_{r \in \{r_{i1}, \dots, r_{iN}\}} v(q_i, r), \qquad \mathcal{S} = \{(q_i, r_i^*) : v(q_i, r_i^*) \geq \tau_v\}, \tag{7.3}$$

随后只在 $\mathcal{S}$ 上对 ϕ 做监督式伪标签更新。VDS-TTT 采用生成候选、验证筛选和 LoRA 更新的路线 [24]。如果 $\mathcal{S}$ 为空，正确实现应跳过更新，而非降低阈值直到有数据可训练。

验证器可能是经过带标签训练的模型、可执行单元测试，或一组形式化约束。它们的覆盖范围不同：通过若干测试不代表程序完全正确，语言模型对答案的高评分也不代表真实正确。把验证器称为“无标签反馈”，只表示当前适应器未直接读取人工答案，并没有抹去验证器训练时的监督和外部工具信息。

评价时将“筛选答案”和“用筛选答案训练”拆开。至少比较固定模型单次回答、固定模型加同一验证器、相同候选预算下再加参数更新。否则收益可能全部来自候选选择。还应检查验证器与生成器的相关错误、重复伪标签、某类题目的系统性漏选，以及更新是否在未来查询上放大错误。将更新限制为 LoRA 降低了可修改维度，并不能保证消除遗忘或行为回归。

7.5 计算预算应与学习收益分别报告

对查询 q，令可观测诊断为 $u(q)$，预算策略 $b(u)$ 决定候选数、推理长度和验证次数。理想的资源分配可写为

$$\max_{b} \ \mathbb{E}[Q(q, b(u(q)))] \quad \text{s.t.} \quad \mathbb{E}[C(q, b(u(q)))] \le B. \tag{7.4}$$

Q 是事后可评价的任务质量，不是部署时已知的答案。若预算策略以真实正确率推断题目难度，则它使用了额外标签，不能作为普通部署策略。测试时计算研究表明预算分配依赖问题和验证机制 [25]；它并不要求跨查询改变参数。

一个有解释力的对照是四组：冻结模型且固定单次预算、冻结模型且增加搜索、参数适应且维持单次回答、适应与搜索同时使用。各组保存生成 token、验证调用、训练 token、前向与反向次数、时延及峰值内存。既报告相同候选数，也报告相同墙钟预算；两者通常不能同时相等。普通 KV cache 在所有可兼容组中使用相同策略，避免把计算复用误记为学习收益。

语言模型的讨论至此回到本书的统一问题：信息从何而来，写入哪个状态，何时能影响评价。下一章将把这些问题放进具有真实约束的应用场景。

第 8 章　应用约束：何时继续、停止与恢复

把自然图像腐蚀上的算法直接推荐给汽车、医院或机器人，会跳过最关键的部署条件。本章将场景改写为待验证的约束分析。以下不构成对任何系统的性能或安全保证；每个候选更新器都需要在相应的数据、时序和代价条件下接受独立评价。

8.1 先填写一份部署约束表

选择损失之前，先回答六个问题：输出多久必须到达；一次能同时看见多少输入；历史能保存多久；哪些状态可写；真实标签何时到达；失败能以什么证据识别。表 8.1 给出可操作的检查点。

表 8.1　场景中的约束、可检验假设与不可省略的对照。表中列的是实验入口，而非方法排名。

场景	约束及偏移	一个可检验假设	关键反证或对照
连续视觉	单帧到达、相关视频帧、天气突变与回访	有限历史统计能改善当前外观偏移	相同帧预算的冻结模型；标签集中、域回访及截止时间
医疗影像	患者与设备成组、患病率变化、稀有病种	预定设备变更下有限更新改善敏感度	患者级独立评价；少数病种、校准与人工复核代价
传感器流	时间相关、时间戳、延迟标签、缺失值	历史状态适应可跟踪指定漂移	严格滚动预测；机制漂移、突变及恢复
机器人	动作改变后续观测、不可任意试错	受限感知更新改善固定策略输入	冻结策略回放、闭环仿真、同初始条件及回滚失败

同一组数据可能包含统计变化、标签先验变化与语义机制变化。无标签诊断不能自动识别它们。若没有可检验假设支撑更新，保留冻结模型并收集诊断，本身就是合理的实验条件。

8.2 自动驾驶：相关帧与截止时间

以连续相机感知为例，某时刻进入隧道，外观迅速变暗，车辆与行人比例也可能变化。若把相邻帧合并成批，统计量同时反映天气、道路结构和当前对象组成。当前批准确率的变化未必来自域适应，也可能来自批内类别比例。

批大小为一并不总使 BatchNorm 无法计算：卷积特征还有空间位置，但空间像素高度相关，有效样本数可能很小；后部全连接特征则可能只有一个观测。改变为 GroupNorm 或 LayerNorm 也不是无需代价的部署开关，替换归一化一般改变原模型计算，可能需要训练或重新验证。应先记录模型实际的归一化层、归约轴与统计来源，再决定可比较的基线。

一个离线压力测试可取固定视频片段，构造日间、夜间、隧道、日间回访的顺序，同时保留类别集中与混合片段。分别运行冻结源模型、历史统计更新和受限参数更新；严格冻结数据 ID 与到达时间。按截止时间计算输出：若更新尚未完成，明确使用旧状态、丢弃更新还是延后输出。把整批最后一帧之后才计算的结果称为逐帧实时预测，会掩盖等待成本。

需要同时画最坏时间窗、对象类别召回、回访恢复与时延分位数。教师平均和源锚模型可以提供对照及恢复点，但教师可能与学生一起犯错，源模型在夜间也可能失效。保留它们不能单独证明系统能够安全运行。本书仅讨论感知适应的时序与评价，完整控制约束属于《具身智能》的主题范围。

8.3 医疗影像：患者独立性与可用概率

医院间差异可能来自扫描仪和重建协议，也可能来自患者构成、诊断标准或疾病谱。将所有差异称为协变量偏移，会把需要重新学习决策关系的问题误当作输入校正。低熵更新尤其可能把少见病灶变得更自信地错误。

首先把患者作为划分单位，同一患者的相邻切片或重复检查不能同时落入独立调参与测试集合。若允许一名患者内适应，须明确是否能读完整扫描、预测是否在全部切片到齐后发生、患者结束后是否重置全部状态。跨患者持续更新则需要额外评估顺序、病种集中和设备回访，不能沿用患者内结果。

一个有边界的实验问题是："在固定设备变更、固定诊断任务和预定更新参数范围下，当前无标签扫描能否改善独立病例的敏感度，同时不增加预定容许范围外的假阳性？"这需要带标签的独立事后评价，并按病例或预先定义的群组给出不确定性，而不是只统计切片数。若标签量不足，不能把稀有病种上未观测到失败解释为风险已排除。

校准、选择性风险和复核率都依赖决策阈值。阈值应在独立开发数据固定；测试标签只用于审计。部署中若获得专家复核标签并用于更新，方法转为带监督或半监督在线适应，需单独报告其反馈延迟和标注成本。单看模型不确定性，无法判定某次适应保持临床可靠性。

8.4 传感器流：因果时间窗与延迟反馈

传感器、语音和时间序列的共同难点是相关性。假设在时间 t 预测未来 h 步的目标，输入窗口为 $x_{t-w+1:t}$，所有统计与更新材料都必须在 t 时可用。用完整序列归一化后再划分滚动窗口，会将未来均值与方差信息送给当前预测。

把状态分为短期估计与持久参数很有帮助：短期状态可跟踪噪声或偏置，持久参数承载跨时段知识。两者应有不同有效期。缓存故障前后的数据并随机打散，会消除时间漂移；按时间评价、按会话或设备分组重采样更贴近问题。

若真实目标 y_t 在 $t+d$ 才到达，可以先以历史无标签信号更新，标签到达后再进入监督队列。报告应分别标记无标签步骤与监督步骤，禁止提前使用 y_t，并记录延迟分布。未来预测残差是较直接的任务反馈，但在标签尚未到达时，只能使用代理诊断；重建损失降低可能仅表示模型更会还原传感器噪声。

突变、渐变和机制漂移应分开测试。例如相同温度读数因设备维护而改变故障含义时，恢复源输入统计未必有效。至少评价漂移前、漂移中、稳定后和旧环境回访四段，并保存缺失值处理和窗口重叠规则。高度重叠的窗口不能当作大量独立样本来缩小置信区间。

8.5 机器人：更新改变了将来会看见什么

离线分类的测试流通常由数据集给定；机器人动作会改变下一步观测。若适应使机器人转向别处，之后看见的数据分布也变化。两套方法在不同轨迹上获得的成功率差异，同时包含模型更新和访问状态的变化，不能当作同一固定目标流上的比较。

可分三步建立证据。先在固定轨迹回放中检查感知预测，隔离数据顺序与更新机制；再在可复位的仿真环境中用相同初始条件和任务分布评估闭环表现；最后才在具体部署约束下研究在线更新。回放表现良好不能替代闭环测试，因为数据分布已被动作改变。

几何一致性、动力学残差和多模态一致性可以作为代理信号，但遮挡、接触变化、传感器共同偏差都可能让它们失效。冻结控制头或保留源策略也不自动维持原约束：感知输入变化已经可能改变动作。需要明确可更新模块、提交更新的时刻、机器人实例间状态隔离，以及恢复触发后的实际行为。更完整的控制接口、安全约束和闭环执行应由具身系统专门设计，本书不以 TTA 机制替代这些工作。

8.6 停止适应是需要验证的决策

把更新门控记为 $g_t \in \{0,1\}$，候选更新 $s'_t = \mathsf{U}(s_t, B_t)$。一个可审计的状态机有四种动作：继续使用冻结状态、提交候选状态、跳过本次更新、恢复预先保存的检查点。阈值与状态转换规则在开发流固定，测试时记录每次触发原因。

可观察诊断包括非有限损失、梯度范数、预测类别直方图、源锚模型分歧、代理损失趋势和时延。只有前两类某些情况可直接判定数值更新无效；其余都是可能误报或漏报的风险提示。例如真实类别集中也会造成类别直方图尖锐，源锚模型分歧也可能来自有益纠正。无标签监测通常不能证明真实任务风险正在下降。

恢复必须同时处理参数、归一化缓冲、教师、优化器、缓存和随机性，且重新计入加载及预热成本。恢复源状态只表示回到已知计算状态，不能保证源模型适合当前环境。第 9 章将把“恢复了哪些状态”和“任务表现是否恢复”转化为不同指标，避免把操作成功与质量恢复混为一谈。

第 9 章　可靠评测：顺序、恢复与累计代价

平均准确率回答“在这些样本上答对了多少”，却没有说明答案用了什么信息、是否按时到达、旧域是否遗忘。本章把指标与协议一起定义。第 5 章的小型实验是其中一种受控实例；不能把其结果外推为真实连续部署结论。

9.1 先冻结实验契约，再运行方法

每次实验保存一份机器可读配置，至少包含下列六组字段。

1. **数据与标签空间：**数据版本、样本 ID、源/目标划分、已知与未知类别、腐蚀生成参数，以及是否能访问源原始样本、源统计或辅助数据。
2. **到达与可见性：**精确顺序文件、批划分、时间戳、当前批统计是否可见、历史窗口长度、域边界是否提供、能否重复访问样本。
3. **预测时刻：**先预测后更新或先更新后预测；增强是否参与计分；当前输入用于多少次前向；异步更新何时提交。
4. **状态生命周期：**参数选择、归一化模式、教师/缓存/优化器/随机数状态；episodic 单元、重置频率、恢复点及缓存淘汰规则。
5. **选择与监督：**超参数、阈值、停止规则在哪些开发数据上确定；测试标签只由谁计分；是否使用延迟标签、验证器、工具或人工反馈。
6. **资源：**模型与实现版本、设备、数值精度、种子、增强数、更新步数、前后向次数、时延、峰值内存及保存状态大小。

这些字段是方法的一部分。把“每域重置”改成“全流持续”，或让当前预测改用适应后参数，都应视为另一实验条件。公开域边界的重置可用作诊断上界，但不能和不知道边界的方法混称相同信息条件。

CIFAR-C、ImageNet-C 一类腐蚀基准有助于控制外观变化 [26]；它们并不覆盖未知类别、设备变化或决策关系变化。现实测试时适应基准强调类别与域的顺序组织 [27, 28]，但引用基准名称不能代替报告实际使用的子集和协议。

9.2 把任务质量、置信度与时间分开

令第 i 个已提交预测为 $p_i \in [0,1]^C$，真实标签为 y_i，$\hat{y}_i = \arg\max_c p_{ic}$。在 n 个样本上的准确率与最近 w 个样本的窗口准确率为

$$A_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{y}_i = y_i\}, \qquad A_{t,w} = \frac{1}{|W_t|}\sum_{i \in W_t} \mathbf{1}\{\hat{y}_i = y_i\}, \tag{9.1}$$

其中 $W_t = \{\max(1, t-w+1), \dots, t\}$。开头不足 w 个样本时分母使用真实窗口长度；窗口长度应预先固定。画 A_n 可看累计表现，画 $A_{t,w}$ 可看短时崩溃，两者不能互相替代。

对预定域集合 $\mathcal{D}$，分别报告各域 A_d、宏平均 $|\mathcal{D}|^{-1}\sum_d A_d$ 和最差域 $\min_d A_d$。总体 A_n 按样本数加权，宏平均按域等权；标签集中或各域规模不同时，两者可明显不同。“最差域”只是在本次预先

指定的域中的最小值，不是对所有未知环境的保证。类别召回同理应标明分母，缺失类别不能按零错误处理。

概率质量可用多分类负对数似然与 Brier 分数：

$$\mathrm{NLL} = -\frac{1}{n}\sum_{i}\log p_{i,y_i}, \qquad \mathrm{Brier} = \frac{1}{n}\sum_{i}\sum_{c=1}^{C}(p_{ic} - \mathbf{1}\{y_i = c\})^2. \tag{9.2}$$

本书 Brier 未再除以 C，比较时须采用同一定义。NLL 用固定且披露的数值截断计算，以避免 $\log 0$；截断值不能根据方法结果调整。

令 $\rho_i = \max_c p_{ic}$，用事先固定的置信度分箱 I_m，则

$$\mathrm{ECE} = \sum_{m:|I_m|>0}\frac{|I_m|}{n}\left|\frac{1}{|I_m|}\sum_{i\in I_m}\mathbf{1}\{\hat{y}_i = y_i\} - \frac{1}{|I_m|}\sum_{i\in I_m}\rho_i\right|. \tag{9.3}$$

ECE 受分箱、样本量与类别构成影响 [29]。给出分箱规则并同时报告 NLL、Brier，比只给 ECE 更容易识别过度自信。熵下降与校准改善没有等价关系；分箱内的正负误差也可能彼此抵消。

若模型允许拒答，对保留集合 $S_\tau = \{i : \rho_i \geq \tau\}$ 定义覆盖率 $|S_\tau|/n$ 与选择性错误率 $|S_\tau|^{-1}\sum_{i\in S_\tau}\mathbf{1}\{\hat{y}_i \neq y_i\}$。空集合的风险记为未定义并报告覆盖率零。不能仅因拒绝大多数难例而称系统准确率提高。开放集还应单独报告未知类检测与已知类分类；已知类高置信度不是未知检测器。

9.3 遗忘、回访与恢复不是同一个数

在不改变适应状态的独立探测集上，令 a_k^{before}、a_k^{after} 是域 k 在某段适应前后的准确率，则有符号性能损失为

$$F = \frac{1}{K}\sum_{k=1}^{K}(a_k^{\mathrm{before}} - a_k^{\mathrm{after}}). \tag{9.4}$$

$F < 0$ 表示平均改善，不应截掉后再仍用同一指标名称。若要报告非负遗忘，另定义 $F_+ = K^{-1}\sum_k \max(0, a_k^{\mathrm{before}} - a_k^{\mathrm{after}})$。探测时应从状态副本运行，禁止探测样本更新主流中的缓存、统计或优化器。

域回访比较的是同一域两次出现的表现，既反映遗忘，也受这两段样本难度影响。应使用匹配样本或重复受控数据，或同时报告各段冻结基线。可定义相对于冻结模型的差额

$$G_d^{(r)} = A_{d,\mathrm{adapt}}^{(r)} - A_{d,\mathrm{frozen}}^{(r)}, \qquad D_d = G_d^{(1)} - G_d^{(2)}, \tag{9.5}$$

其中 $r = 1, 2$ 为首次与回访段。$D_d > 0$ 表示相对冻结基线的优势减弱；由于两段样本不同，它仍不是纯粹的因果遗忘估计。

恢复至少有两层含义。状态恢复先用文件哈希确认检查点身份，再核对载入后的完整状态与保存状态一致；文件身份本身不能证明载入操作完整；任务恢复检查回访后的准确率或损失是否回到预定水平。令域回访从 t_0 开始，取固定参考准确率 a_{ref}、容差 ϵ 和连续窗口数 K_0，可将恢复延迟定义为首个满足连续 K_0 个完整窗口均有 $A_{t,w} \geq a_{\mathrm{ref}} - \epsilon$ 的时间点减去 t_0。若流结束仍未满足，标记“未恢复”并报告观察长度，不能填零。

这个恢复指标用真实标签进行事后测量，不是部署时可直接使用的触发器。实际系统的触发器若只访问无标签信号，须另报其误报、漏报、回滚次数和加载代价。预设域边界后立即重置的结果，也不能充当“自动检测并恢复”的证据。

9.4 四种顺序如何构成可解释的压力测试

固定基础样本及扰动版本，仅改变到达顺序，分别建立：

1. **原序：**保留数据集或采集过程的顺序，披露其是否具有时间意义；文件名顺序不自动等于真实时间顺序。
2. **随机打乱：**按固定种子排列相同样本，保存完整 ID 列表，避免只保存种子却无法重建。
3. **类别集中：**由离线评估器按真实类别组织流；类别用于构造压力条件，不发送给适应器，也不用于平衡缓存。
4. **域回访：**构造 A–B–A 段并披露回访是相同样本、不同样本还是重复腐蚀。重复输入带来的记忆收益须与域泛化区分。

严格比较顺序效应时，每条流使用相同样本多重集与评价权重。若 A–B–A 为得到回访而重复 A，应在其他顺序也保留同一多重集；否则同时改变了顺序与曝光次数。为隔离批组成和跨批状态效应，可以分别固定批内组成只打乱批顺序，以及重新划分批。两种试验回答不同问题。

每种顺序均报告持续状态与固定单元重置。冻结模型对纯逐样本前向的顺序应不敏感，这可作为独立参考检查；使用当前批统计的模型即使没有梯度更新，也可能因批组成而改变结果。不能把这类差异全归因于优化器。

9.5 累计代价不能只写成每步耗时

对第 t 个单元记录前向数 f_t、反向数 b_t、增强数 a_t、额外训练 token 或图像数，以及真实时间 Δt_t。累计计算记录可写为

$$C_T^{\mathrm{time}} = \sum_{t=1}^{T} \Delta t_t, \quad C_T^{\mathrm{fwd}} = \sum_{t} f_t, \quad C_T^{\mathrm{bwd}} = \sum_{t} b_t, \quad L_T = \sum_{i=1}^{n_T} \ell(p_i, y_i). \tag{9.6}$$

L_T 是累计任务损失，不能用来反向调测试超参数。时间测量须说明预热、设备同步、数据加载、增强、恢复、验证器和缓存管理是否计入。异步方法还要报告未完成更新数、提交滞后与丢弃率。平均时延、尾部时延和吞吐量分别反映不同约束。

公平比较可以有多个资源约束视角：固定墙钟预算、固定前后向预算、固定峰值内存。它们不是相同约束，最好画任务质量对实际累计代价的曲线。若把错误与时间写成单一目标 $L_T + \lambda C_T^{\mathrm{time}}$，必须说明 λ 的单位与实际决策含义；不能把毫秒和错误率随意相加。没有测量设备功耗时，不把运算次数换称能耗。

9.6 消融应能排除一个解释

表 9.1 围绕机制设计消融。每项只改变待检验因素，其他信息与预算固定。

待排除的解释	对照	需要观察什么
提升只来自当前批统计	冻结源统计、仅批统计、批统计加梯度	三者预测时刻一致，区分统计与参数收益
提升只来自增强集成	相同增强、无更新的集成	同等前向数下更新是否仍有增益
筛选只减少训练量	匹配样本数的随机筛选	接受率、类别分布、累计反向数
教师只是复制历史	无教师、等预算冻结锚、EMA 教师	相同初始化下伪标签误差和长期退化
恢复依赖额外域边界	无边界检测、给定边界、定频重置	误报、漏报、恢复延迟及回访结果
缓存只是扩大曝光	无缓存、同容量随机缓存、目标缓存	命中率、年龄、错误条目与每样本访问次数
方法只在易顺序有效	同一多重集的四类顺序	最坏窗口、校准、回访与累计代价

超参数使用与最终测试分开的开发域或开发流确定，并公开选择准则。多次尝试顺序、种子、增强和停止时机后只报告最优一次，同样产生选择偏差。对高度相关流，逐样本独立 bootstrap 会夸大有效样本量；可用独立流、多次完整种子运行，或按会话/设备/时间块重采样，并披露块长度。只有一个受控流时，就如实报告这个流的结果，不伪造总体置信结论。

第 10 章　从失败模式到可检验的研究问题

一个新损失函数不是研究设计的起点。更有价值的起点是：现有方法在明确的信息和时序条件下发生了什么可重复的失败，替代解释是什么，怎样的结果会推翻改进假设。本章将原稿中的方向列表改成能够执行并可能失败的实验设计。

10.1 先定义“不更新”的价值

设门控 $g_\psi(s_t, B_t) \in \{0,1\}$ 决定是否提交更新，$g = 0$ 保持当前状态。门控可以读取归一化统计、预测分布、梯度和历史诊断，但不能读取用于计分的当前标签。真正希望比较的是长期任务质量，而不是门控接受率或代理损失本身。

一个评价目标可写为

$$J(\psi) = \mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathsf{F}(s_t, B_t), Y_t) + \lambda c_t + \gamma \kappa_t\right], \tag{10.1}$$

其中 c_t 是定义清楚的资源代价，κ_t 是恢复事件或其他事先定义的损失；λ, γ 将它们换到同一决策尺度。状态按预测与更新约定递推。式中真实标签用于离线开发与事后评价，不能据此让部署门控偷看 Y_t。若用有标签开发流训练 ψ，应明确它是受监督训练、部署时无标签运行的门控。

预期“熵低时才更新”有效，需要至少两个假设：源预测的低熵与正确性有可迁移联系，且被接受样本的梯度对后续数据有帮助。第 2 章已经给出前一个假设的反例。一个完整实验应包括高置信错误、真实类别集中与开放集输入，允许结论是门控在这些情形下没有作用。

10.2 一个可以完整执行的门控研究

10.2.1 问题与对照

研究问题限定为：“在固定图像模型、当前批统计和先预测后更新协议下，低熵筛选能否在同等训练量下改善连续流表现？”更新对象固定为归一化 affine，基础损失固定为条件熵，避免同时改变模型容量与反馈信号。

使用三个对照：每批都更新；低熵阈值筛选；匹配接受数量的随机筛选。冻结源模型与仅统计更新作为共同基线。随机对照按批匹配接受数量时，要说明它获得了门控产生的计数信息，这只是诊断消融；独立可部署的随机策略应在开发流固定其概率。若低熵方法只因少做反向而更稳定，匹配计算量的随机筛选可能获得相同结果，从而推翻“筛选了更可靠样本”的解释。

10.2.2 划分、失败条件与记录

将开发扰动参数、样本 ID 与测试流分开，阈值只在开发流确定。固定第 9 章四种顺序与预算，保存每次接受/拒绝、损失、类别直方图、梯度范数、状态哈希和预测。标签由独立评估器事后补入日志，适应代码只接收图像。

事先规定主要终点，例如按流等权的准确率与最坏窗口准确率；ECE、接受率和累计时间是辅助指标。若主要终点没有改善，不能改用表现最好的辅助指标来宣称总体成功。若效应只存在于某一个阈值、某一个类别顺序或额外计算量，结果应收窄到这些条件，并给出失败样本的诊断。

这项研究还需要反事实边界：单步梯度有益不表示长期提交有益；门控拒绝的样本若不再被学习，后续状态轨迹也改变。可用短期分叉状态副本比较单次提交与不提交的未来窗口，但该诊断不得用未来标签控制真实运行轨迹。记录分叉长度和额外成本，它只解释机制，不能冒充部署算法。

10.3 四个有明确边界的扩展方向

1. **记忆与参数容量**。在相同存储字节、读写预算和输入曝光下比较原型缓存、文本记忆与低秩增量。区分单次查询收益、跨查询迁移和污染持续时间。若缓存因保存原样本获得更多信息，需另列信息条件，不能解释为表示形式本身优越。
2. **统计偏移与机制变化**。构造外观变化但标签规则固定，以及输入相似但标签规则改变的两类受控流。验证同一诊断能否区分二者；若无标签输入边际相同，则不可辨识性会限制任何门控的保证。研究目标可以是识别可适应的子集，而非承诺普遍识别。
3. **跨模态反馈**。比较单模态、互相独立的噪声、以及共同偏差下的一致性更新。两模态一致可能一起错误，因而需要已知扰动机制和独立任务标签作事后评价。删去一模态和固定融合权重可排除单纯集成收益。
4. **资源受限与黑盒接口**。只允许 logits、embedding 或固定低精度前向时，比较输出校正、缓存和可部署的无梯度候选搜索。将调用数、可见输出精度与状态大小固定，检查改善是否来自额外模型调用或接口信息。没有真实设备测量时，仅报告算法代价，不声称硬件节能。

校准门控、保形预测或其他不确定性工具可以成为这些实验的组成部分，但其有限样本结论依赖交换性、数据划分或漂移假设。连续流经过自适应更新后，这些条件不自动保持。若要提出覆盖率或风险保证，必须写出概率空间、保证作用的对象及假设，并验证部署协议满足它们；仅在方法名称中加入“可靠”并没有完成证明。

10.4 实验包应让别人重建状态轨迹

一个可复用研究包应包含配置、数据构造、顺序 ID、初始模型、软件版本、全部种子、逐步预测与更新日志，以及从干净环境开始的运行命令。对不能分发的数据，提供下载或生成规则与哈希。对不能执行的大模型方案，交付明确协议、资源估计和待验证假设，结果表标为未运行。

撰写结果时将四类证据分开：手算反例证明某种推论不成立；受控合成流展示机制；公开基准实测量化限定条件下的表现；论文报告只是对应论文条件下的外部证据。任何一类都不自动替代另一类。负面结果尤其有用：若同样本同预算的冻结或统计基线已经解释全部提升，就应保留这个结论，而不是再扩大搜索以追求一个正数。

本书的实践目标是能把一个适应系统描述成完整的实验对象：它看见什么、写入什么、在何时输出、为何可能失败，以及失败后能否恢复到已知状态。将这些问题回答清楚，才有依据决定下一次更新是否值得进行。

附录 A　术语、符号与练习提示

A.1　术语对照

表 A.1　本书术语表。英文缩写相同不代表实验条件相同。

中文	英文或缩写	本书采用的含义与边界
测试时训练	Test-Time Training, TTT	经典路线在源训练期准备辅助任务，测试时自监督更新；用于架构内学习时另说明快速权重。原题“更新时训练”为误译，全文采用此译名。
测试时适应	Test-Time Adaptation, TTA	部署期利用输入及允许的信息调整预测系统，包括参数、统计、提示或其他状态。
完全测试时适应	Fully TTA / FTTA	适应阶段不访问源原始训练数据；源模型和允许保存的统计仍含源信息。
连续测试时适应	Continual TTA, CTTA	状态跨到达单元保留，环境可随时间改变；不隐含可访问域边界。
回合式重置	Episodic reset	每个指定单元从相同完整初始状态开始；单元大小须披露。
混合域	Mixed-domain	同批或短窗含不同环境；是数据组织条件，与重置策略独立。
先预测后更新	Predict-then-update	先固定当前输出，再用当前输入更新后续状态；统计可见性另报。
先更新后预测	Update-then-predict	当前输入先参与无标签更新，再以更新后状态评价当前输入。
源数据不可访问适应	Source-Free Domain Adaptation, SFDA	源模型可用而源数据不可用；目标数据可为离线集合，不必严格在线。
上下文学习	In-Context Learning, ICL	通过示例和上下文改变条件预测；全局参数无须更新，示例可包含监督。
测试时计算	Test-Time Compute, TTC	增加采样、搜索、推理或验证预算；本身不等于模型适应。
快速权重	Fast weights	在序列或任务内按内层学习规则写入的临时参数状态，区别于训练后固定的慢速权重。
键值缓存	Key–value cache / KV cache	保存已处理 token 的键值表示以复用计算；普通缓存不自动产生新学习信号。
教师平均	Exponential Moving Average, EMA	用指数滑动平均平滑参数；不是独立标签或正确性保证。
伪标签	Pseudo-label	模型或验证器产生的训练目标；错误可被后续更新放大。
条件熵	Conditional entropy	对每个输入预测分布求熵，再跨输入平均；降低它鼓励自信。
增强边际熵	Marginal entropy over views	同一输入多个视图预测均值的熵；与跨样本类别多样性目标不同。

中文	英文或缩写	本书采用的含义与边界
校准	Calibration	预测概率与实际正确频率的关系；低熵不等于校准好。
标签先验变化	Label-prior change	类别比例变化；只有再满足类条件分布不变时才是经典标签偏移假设。
开放集	Open set	目标可能出现源标签空间外的类别，需要与已知类分类分开评价。
恢复 / 回滚	Recovery / rollback	前者可指任务质量恢复，后者是恢复检查点操作；操作完成不代表质量已恢复。
带监督在线学习	Supervised online learning	使用当前或延迟真实标签更新，需披露反馈延迟，不能称为纯无标签适应。

A.2 符号与记录单位

表 A.2 主要符号。下标 t 在部署章节表示到达单元，j 在语言模型章节表示序列位置。

符号	含义	单位或约定
P_s, P_t	源联合分布、时刻 t 的目标联合分布	固定目标域时可不随时间变
R_s, R_t	给定任务损失下的源风险、目标风险	与损失函数一致
θ_0, θ_t	部署初始参数、当前参数	初始检查点应保存哈希
s_t	第 t 个到达单元前的完整系统状态	参数、缓冲、教师、记忆及优化器等
B_t, Y_t	当前输入批、仅评估器可用的真实标签	批大小与有效样本数不同
$\mathcal{I}_t$	更新时允许访问的信息集合	无标签协议排除 Y_t
F, U	预测算子、状态更新算子	需连同时序约定定义
$b_t, \bar{\theta}_t$	归一化缓冲、教师参数	b_t 不是 BN affine 参数
$\phi_t, \mathcal{M}_t$	提示/低秩增量、记忆或缓存	保存范围与有效期须固定
o_t, r_t	状态定义中的优化器、随机数状态	按完整随机数状态保存
p_c, z_c, C	类别概率、logit、类别数	概率和为 1；C 为无量纲整数
H, D_{KL}	熵、KL 散度	自然对数，单位 nat
$\bar{p}$	指定视图或样本集合中的预测均值	必须说明平均的集合
η, λ, τ	学习率、目标权重、阈值	含义依所在目标重新说明
γ, β	归一化 affine 缩放、平移	并非所有模型都使用同名参数
W_j	第 j 次写入后的快速权重	不等于全局模型参数
L_{in}, PPL	输入平均 NLL、其指数	token 掩码与平均方式固定
$A_n, A_{t,w}, A_d$	累计、窗口、域准确率	以 $[0,1]$ 或百分比报告并注明
F, F_+	有符号性能损失、非负遗忘	按独立探测域等权
$G_d^{(r)}, D_d$	第 r 段相对冻结基线优势、回访优势损失	见式 (9.5)
C_T^{time}	累计墙钟时间	秒，说明是否含加载与恢复
$C_T^{\mathrm{fwd}}, C_T^{\mathrm{bwd}}$	累计前向与反向次数	输入大小与增强数同时报告

同一字母在不同文献中常有不同含义。本书核心状态采用式 (1.2)；局部公式仍需逐处核对定义，不应仅凭字母相同推断对象相同。实现记录宜使用有语义的字段名，例如 `optimizer_state` 与 `rollback_count`，避免把数学简写直接变成含糊日志。

A.3 练习：从推导到协议审查

以下练习既可手算，也可作为代码审查任务。提示提供解题路线与可独立检查的结果，不预设任何方法必然提高准确率。

1. **熵、错误与温度**。从 $p = \sigma(z)$ 推导 $\mathrm{d}H/\mathrm{d}z$；令真实标签为 0、$z > 0$，证明足够小的正学习率降低熵但增加真实 NLL。再说明温度缩小为什么能在准确率不变时改变熵。
   提示：利用 $\log[p/(1-p)] = z$ 得到 $-zp(1-p)$。梯度下降使正 z 增大，而错误标签的 NLL 为 $\log(1+e^z)$，对 z 的导数为 $p > 0$。正温度不改变 logit 大小顺序。这里没有声称准确率一定下降，因为一个已经错误的样本可能始终错误。
2. **边际熵的两种角色**。证明式 (2.7)，给出两个视图各自熵为零但边际熵非零的例子；再解释为什么跨样本信息最大化目标对单类别批可能不合适。
   提示：取 $p_1 = (1,0)$、$p_2 = (0,1)$，平均熵为零、均值熵为 $\log 2$。若两张图确实都是同一类别，多样性项反而偏好把某些预测推到别的类别；它需要真实类别先验的支持。不要把“同输入视图一致”与“不同输入类别多样”混为同一要求。
3. **一步更新的有效范围**。由式 (2.2) 求保证主任务损失上界下降的步长区间；当梯度内积为零或为负时，能否从该上界推出严格下降？
   提示：在 $g_a \neq 0$、内积为正时解一元二次不等式；区间为 $0 < \eta < 2\langle g_m, g_a\rangle/(L\|g_a\|^2)$。内积不为正时，这个界不给出严格下降保证，并不单凭界的松弛性证明实际损失一定上升。无标签适应器还无法直接计算 g_m。
4. **完整重置**。一段程序只载入模型参数，保留 Adam 动量、BN 缓冲和缓存。设计一个独立检查，判断两次 episodic 运行是否从同一状态开始。
   提示：保存完整状态副本，构造固定输入和固定增强随机数，比较预测、下一次更新后的全部张量及缓存内容。先运行一个不同历史批再重置，检查两条轨迹是否一致。模型 `state_dict` 通常含缓冲却不含优化器；仅比较参数哈希不足以证明重置完整。
5. **顺序效应的来源**。固定同一组图像，为什么冻结模型、仅当前批统计模型、连续 Tent 风格模型可能有不同的顺序敏感性？设计两个试验区分批组成与跨批参数历史。
   提示：先固定每批成员、只打乱批顺序，再固定总样本、重新组批。确定性逐样本冻结模型的样本预测应相同；批统计模型在第二种试验中可变化；持续参数更新在第一种试验中也可变化。若存在随机增强，先固定每样本随机性，否则多了一个解释。
6. **语言模型信息审计**。一个系统读入含参考答案的文档，以自回归损失更新 LoRA，再回答同一题；另一个系统只增加候选数并用同一验证器筛选。分别说明它们修改了什么、使用了什么监督，如何构造公平对照。
   提示：前者更新参数且访问答案，不能称为无标签任务泛化；后者可保持参数冻结，但验证器仍有额外信息。去除参考答案、固定训练文本掩码，加入不更新但读取相同上下文的模型；同时固定候选和验证预算，再比较参数更新的边际收益。
7. **恢复指标的反例**。模型成功载入源检查点后，在回访域仍未达到预定准确率。证明“状态回滚成功”不足以推出“任务恢复成功”，并说明恢复延迟应如何记录。

提示：源模型本来就可能不适合回访域。可构造源检查点在该域准确率低于阈值的固定例子。采用第 9 章的连续窗口规则，若观察结束未恢复，记录为未恢复及观察长度；不能以文件读取时间替代任务恢复时间。

8. **挑选一个会失败的研究假设**。从“低熵意味着可靠”“类别直方图越均匀越好”“教师比学生正确”中选一个，构造受控反例，再提出不会偷看测试标签的诊断实验。
   提示：分别可用高置信错误、真实单类别流、师生共享系统性偏差。诊断信号可以测量，正确性评价仍需独立标签。一个有效答案应区分门控可见字段与评估器字段，预设失败判据，并允许实验结论是否定原假设。

A.4 按已有基础安排实践

先用第 2 章的手算与独立数值检查确认损失含义，再复现第 5 章冻结、统计、梯度三组对照。每新增教师、筛选或缓存，只增加一个组件并保留相同流。最后把同一信息审计表用于 VLM 或 LLM，先比较状态和预算，再决定是否需要昂贵实验。

附录练习不要求复现原稿列出的全部方法。能独立重建一条状态轨迹、识别一次标签泄漏、解释一个负面结果，比未经核查地拼接许多缩写更接近本书的学习目标。

参考文献

[1] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt. "Test-Time Training with Self-Supervision for Generalization under Distribution Shifts". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9229–9248. arXiv: 1909.13231v3. URL: https://proceedings.mlr.press/v119/sun20b.html (visited on 10/22/2026).

[2] Y. Gandelsman, Y. Sun, X. Chen, and A. A. Efros. "Test-Time Training with Masked Autoencoders". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 29374–29385. DOI: 10.52202/068431-2130. arXiv: 2209.07522v1. URL: https://papers.nips.cc/paper_files/paper/2022/hash/bcdec1c2d60f94a93b6e36f937aa0530-Abstract-Conference.html (visited on 10/22/2026).

[3] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. "Tent: Fully Test-time Adaptation by Entropy Minimization". In: *International Conference on Learning Representations*. 2021. arXiv: 2006.10726v3. URL: https://arxiv.org/abs/2006.10726v3 (visited on 10/22/2026).

[4] M. Zhang, S. Levine, and C. Finn. "MEMO: Test Time Robustness via Adaptation and Augmentation". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 38629–38642. DOI: 10.52202/068431-2799. arXiv: 2110.09506v3. URL: https://papers.nips.cc/paper_files/paper/2022/hash/fc28053a08f59fccb48b11f2e31e81c7-Abstract-Conference.html (visited on 10/22/2026).

[5] M. Shu, W. Nie, D.-A. Huang, Z. Yu, T. Goldstein, A. Anandkumar, and C. Xiao. "Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 14274–14289. DOI: 10.52202/068431-1038. arXiv: 2209.07511v1. URL: https://papers.nips.cc/paper_files/paper/2022/hash/5bf2b802e24106064dc547ae9283bb0c-Abstract-Conference.html (visited on 10/22/2026).

[6] Y. Wu and K. He. "Group Normalization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 3–19. arXiv: 1803.08494v3. URL: https://openaccess.thecvf.com/content_ECCV_2018/html/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.html (visited on 10/22/2026).

[7] J. L. Ba, J. R. Kiros, and G. E. Hinton. *Layer Normalization*. Version arXiv v1. arXiv. 2016. arXiv: 1607.06450v1. URL: https://arxiv.org/abs/1607.06450v1 (visited on 10/22/2026).

[8] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. *Revisiting Batch Normalization For Practical Domain Adaptation*. Version arXiv v4. arXiv. 2016. arXiv: 1603.04779v4. URL: https://arxiv.org/abs/1603.04779v4 (visited on 10/22/2026).

[9] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. "Efficient Test-Time Model Adaptation without Forgetting". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 16888–16905. arXiv: 2204.02610v2. URL: https://proceedings.mlr.press/v162/niu22a.html (visited on 10/22/2026).

[10] S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan. "Towards Stable Test-Time Adaptation in Dynamic Wild World". In: *International Conference on Learning Representations*. 2023. arXiv: 2302.12400v1. URL: https://arxiv.org/abs/2302.12400v1 (visited on 10/22/2026).

[11] Q. Wang, O. Fink, L. Van Gool, and D. Dai. "Continual Test-Time Domain Adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 7201–7211. arXiv: 2203.13591v1. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Continual_Test-Time_Domain_Adaptation_CVPR_2022_paper.html (visited on 10/22/2026).

[12] L. Yuan, B. Xie, and S. Li. "Robust Test-Time Adaptation in Dynamic Scenarios". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 15922–15932. arXiv: 2303.13899v1. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Yuan_Robust_Test-Time_Adaptation_in_Dynamic_Scenarios_CVPR_2023_paper.html (visited on 10/22/2026).

[13] M. Boudiaf, R. Mueller, I. Ben Ayed, and L. Bertinetto. “Parameter-free Online Test-time Adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 8344–8353. arXiv: 2201.05718v2. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Boudiaf_Parameter-Free_Online_Test-Time_Adaptation_CVPR_2022_paper.html (visited on 10/22/2026).

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. arXiv: 2103.00020v1. URL: https://proceedings.mlr.press/v139/radford21a.html (visited on 10/22/2026).

[15] R. Imam, H. Gani, M. Huzaifa, and K. Nandakumar. *Test-Time Low Rank Adaptation via Confidence Maximization for Zero-Shot Generalization of Vision-Language Models*. Version arXiv v1. arXiv. 2024. arXiv: 2407.15913v1. URL: https://arxiv.org/abs/2407.15913v1 (visited on 10/22/2026).

[16] J. Hassan, H. Gani, N. Hussein, M. U. Khattak, M. Naseer, F. S. Khan, and S. Khan. “Align Your Prompts: Test-Time Prompting with Distribution Alignment for Zero-Shot Generalization”. In: *Advances in Neural Information Processing Systems*. 2023. arXiv: 2311.01459v2. URL: https://arxiv.org/abs/2311.01459v2 (visited on 10/22/2026).

[17] H. S. Yoon, E. Yoon, J. T. J. Tee, M. Hasegawa-Johnson, Y. Li, and C. D. Yoo. “C-TPT: Calibrated Test-Time Prompt Tuning for Vision-Language Models via Text Feature Dispersion”. In: *International Conference on Learning Representations*. 2024. arXiv: 2403.14119v3. URL: https://arxiv.org/abs/2403.14119v3 (visited on 10/22/2026).

[18] A. Karmanov, D. Guan, S. Lu, A. El Saddik, and E. Xing. “Efficient Test-Time Adaptation of Vision-Language Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 14162–14171. arXiv: 2403.18293v1. URL: https://openaccess.thecvf.com/content/CVPR2024/html/Karmanov_Efficient_Test-Time_Adaptation_of_Vision-Language_Models_CVPR_2024_paper.html (visited on 10/22/2026).

[19] X. Hu, K. Zha, M. Sun, A. Chen, C.-H. Kuo, and R. Nevatia. *BaFTA: Backprop-Free Test-Time Adaptation For Zero-Shot Vision-Language Models*. Version arXiv v2. arXiv. 2024. arXiv: 2406.11309v2. URL: https://arxiv.org/abs/2406.11309v2 (visited on 10/22/2026).

[20] M. Suzgun, M. Yuksekgonul, F. Bianchi, D. Jurafsky, and J. Zou. *Dynamic Cheatsheet: Test-Time Learning with Adaptive Memory*. Version arXiv v1. arXiv. 2025. arXiv: 2504.07952v1. URL: https://arxiv.org/abs/2504.07952v1 (visited on 10/22/2026).

[21] J. Hu, Z. Zhang, G. Chen, X. Wen, C. Shuai, W. Luo, B. Xiao, Y. Li, and M. Tan. “Test-Time Learning for Large Language Models”. In: *Proceedings of the 42nd International Conference on Machine Learning*. Vol. 267. Proceedings of Machine Learning Research. PMLR, 2025, pp. 24823–24849. arXiv: 2505.20633v1. URL: https://proceedings.mlr.press/v267/hu25z.html (visited on 10/22/2026).

[22] T. Zhang, S. Bi, Y. Hong, K. Zhang, F. Luan, S. Yang, K. Sunkavalli, W. T. Freeman, and H. Tan. *Test-Time Training Done Right*. Version arXiv v1. arXiv. 2025. arXiv: 2505.23884v1. URL: https://arxiv.org/abs/2505.23884v1 (visited on 10/22/2026).

[23] A. Tandon, K. Dalal, X. Li, D. Koceja, M. Rød, S. Buchanan, X. Wang, J. Leskovec, S. Koyejo, T. Hashimoto, C. Guestrin, J. McCaleb, Y. Choi, and Y. Sun. *End-to-End Test-Time Training for Long Context*. Version arXiv v2. arXiv. 2025. arXiv: 2512.23675v2. URL: https://arxiv.org/abs/2512.23675v2 (visited on 10/22/2026).

[24] M. M. Moradi, H. Amer, S. Mudur, W. Zhang, Y. Liu, and W. Ahmed. *Continuous Self-Improvement of Large Language Models by Test-time Training with Verifier-Driven Sample Selection*. Version arXiv v2. arXiv. 2025. arXiv: 2505.19475v2. URL: https://arxiv.org/abs/2505.19475v2 (visited on 10/22/2026).

[25] C. Snell, J. Lee, K. Xu, and A. Kumar. *Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters*. Version arXiv v1. arXiv. 2024. arXiv: 2408.03314v1. URL: https://arxiv.org/abs/2408.03314v1 (visited on 10/22/2026).

[26] D. Hendrycks and T. Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *International Conference on Learning Representations*. 2019. arXiv: 1903.12261v1. URL: https://arxiv.org/abs/1903.12261v1 (visited on 10/22/2026).

[27] Y. Yu, L. Sheng, R. He, and J. Liang. *Benchmarking Test-Time Adaptation against Distribution Shifts in Image Classification*. Version arXiv v1. arXiv. 2023. arXiv: 2307.03133v1. URL: https://arxiv.org/abs/2307.03133v1 (visited on 10/22/2026).

[28] C. Du, Y. Wang, J. Guo, Y. Han, J. Zhou, and G. Huang. *UniTTA: Unified Benchmark and Versatile Framework Towards Realistic Test-Time Adaptation*. Version arXiv v1. arXiv. 2024. arXiv: 2407.20080v1. URL: https://arxiv.org/abs/2407.20080v1 (visited on 10/22/2026).

[29] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1321–1330. arXiv: 1706.04599v2. URL: https://proceedings.mlr.press/v70/guo17a.html (visited on 10/22/2026).