Reward-based Continuous Pre-Training

1. 动机：NTP 的短视、CE 的稀疏奖励，以及 RL 在推理上的局限

1.1 NTP 的结构性问题：Pitfalls of NTP & MTP/FSP

Bachmann & Nagarajan 在 ***The Pitfalls of Next-Token Prediction*** 中区分了“teacher-forced 训练”和“自回归推理”两个阶段，指出很多讨论只关注“误差在推理阶段会累积”，但忽视了一个更根本的问题：**在某些规划类任务上，teacher forcing 本身就学不到正确的 next-token predictor**。他们构造了一个极简 planning 任务（包括图搜索、算术等），在这个任务上 Transformer/Mamba 都在 teacher-forcing 下失败，而简单的 multi-token objective 能显著缓解这一 failure mode。

这篇文章给出两个关键点：

- NTP **结构性地偏好短视局部模式**，很难在 teacher-forcing 下“反向”从未来学习到规划；
- Multi-Token Prediction（MTP）在这些任务上能明显修复 NTP 的失败——从损失上看，本质是给模型看更长 horizon 的目标。

Nagarajan 等人在 ***Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction*** 中进一步从“开放式算法创造力”角度提出：在需要“隐式随机规划”与构造新模式的任务上（抽象知识图探索、模式构造等），NTP 训练出来的模型表现出明显的**记忆化与近视性**，而 teacherless 训练、diffusion 等 multi-token 方法在生成多样性与创造性上显著更好。

1.2 Future Summary Prediction：把“规划”显式塞进预训练目标

Mahajan 等人在 ***Beyond Multi-Token Prediction: Pretraining LLMs with Future Summaries*** 中提出了 **Future Summary Prediction (FSP)**：他们认为在MTP 假设在给定前缀的时，预测的 token 是独立的，**导致对未来较长跨度的真实联合分布的近似值较**

差。**于是**在标准 NTP 的基础上，加一个辅助 head，在每个位置 t 预测一段**未来摘要** F_t，希望 LM 的隐藏状态能保留与长程生成相关的信息。FSP 提供了两种 summary：

- 手工的 bag-of-words summary（例如未来窗口内出现过的 token 集合）；
- learned summary：用一个 **Reverse LM**（从右到左）对未来 token 编码得到 embedding。

他们在长文本生成与规划相关任务上，展示了 FSP 对长程一致性和规划质量的提升，说明**在预训练阶段显式引入“未来信息”约束，确实有助于长程能力，缓解 teacher forcing 训练带来的问题**。

> This design reduces teacher forcing by predicting x_{t+k} from $x \leq t$ only, rather than conditioning on the full prefix $x \leq t+k-1$. This is the key principle behind multi-token prediction, i.e., reduced teacher forcing by requiring the model to predict a block of future tokens at each step.

我们在之前的讨论里也吐槽过 FSP 的一些问题：

- 训练一个 ReLM 感觉有点鸡肋，而且右到左的预测对语言建模的 inductive bias 有些怪；
- FSP 直接用回归拉近 LM 与 ReLM 的表征，会有“表示同形性”问题（两个模型的 feature space 是否可比？）。

这也促使我们去设计**更自然、更 CPT 友好的 future encoder**。

1.3 Thinking Tokens：互信息峰值与“思考 token”

Qian 等人在 ***Demystifying Reasoning Dynamics with Mutual Information: Thinking Tokens are Information Peaks in LLM Reasoning*** 中，从信息论角度分析了大模型推理过程：在生成过程中追踪“中间表征与正确答案之间的互信息 $I(h_t, y)$”的变化，发现存在明显的 **MI 峰值（information peaks）**——在特定生成步 MI 会突然飙升，这些 token 往往就是 “Hmm”, “Wait”, “Therefore” 等思考/转折词（他们称为 **thinking tokens**）。

他们还给出理论分析：MI 越高，预测错误概率越低；并实证表明加/删这些 thinking tokens 会显著影响推理性能。因此：

- MI 峰值可以看作**推理路径中的关节点**;
- MI 本身是一个自然的信号，可以用来**监控和塑造推理过程**。

这为我们后面“用 MI 作为 reward shaping 信号”提供了理论支撑。

1.4 RLVR 与 RLP：RL 在“后训练”阶段的作用 & 限制

Yue 等人在 ***Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?*** 中系统分析了 RLVR 在数学、编程等任务上的效果。结论非常关键:

- 在小的 k（如 pass@1）上，RL 模型确实优于 base 模型;
- 但在大的 k（pass@k）上，**base 模型可以通过采样足够多条路径达到类似甚至更高的性能**;
- 分析 reasoning path 发现：RL 模型的“新推理路径”几乎都已经在 base 模型的采样分布中存在——RL 只是 **reweight 这些路径，让高 reward 轨迹更容易被采样到**;
- Distillation 反而更可能引入“新的知识/模式”。

这说明:

> 传统 post-training RL（尤其是 RLVR）没有根本性地拓展推理边界，而是在 base 能力边界内做权重重排。

与之呼应，MARA的 ***Reinforcement Pre-Training*** 以及NVIDIA 的 ***RLP: Reinforcement as a Pretraining Objective*** 工作把 RL 提前到预训练阶段，把一段短 CoT 看作动作，RLP 使用“对下一个 token 的信息增益”作为奖励，让模型在生成下一个 token 之前仔细思考，从而在 CPT 阶段奖励有用的思考过程。

但是:

- RLP 仍然依赖 **显式 rollout 的 CoT**，训练成本高、pipeline 复杂;
- reward 依然围绕 next-token CE 搭建，本质仍然“服务于 NTP”。

1.5 AvataRL：在 pre-training 中使用 reward-weighted log-prob 的示范

AvataRL 把预训练彻底重写为一个 RL 问题：student LM 在每个状态 s_t上，从自己的 top-k 和 critic 模型的 top-k 加上 ground truth 中构造一个动作集合A_t，再根据 reality expert（ground truth + label smoothing）与 critic expert 的几何平均得到一个“理想分布” $p_{\text{ideal}}(a \mid s_t)$，利用该分布的概率作为奖励 $r(a)$，最终优化的是：

$$\mathcal{L}_{\text{AvataRL}}(\theta) = -\sum_{t=1} \left[\sum_{a \in A_t} r(a) \log \pi_\theta(a|s_t) + \beta H(\pi_\theta(\cdot|s_t)) \right]$$

$$\mathcal{L}_{\text{CE}}(\theta) = -\sum_t \log \pi_\theta(y_t|s_t) = -\sum_t \sum_v \mathbf{1}[v = y_t] \log \pi_\theta(v|s_t)$$

从形式上看，这就是一个**reward-weighted log-prob**：reward 是 dense 的（所有激活 token 都有分数，传统的交叉熵训练只有 ground truth token 才有得分，其余 token 得分都是 0），训练仍然基于 teacher forcing 的上下文，兼顾了 RL 的表达力和 NTP 的高效并行。

但是：

- AvataRL 中仍然依赖了一个预训练好的 critic model，这本质上还是对 teacher model 的蒸馏。

这给我们一个重要模板：

> 在 不牺牲 teacher-forcing 并行性 的前提下，可以引入 token-level reward landscape，对 logits 做 reward shaping。

1.6 问题陈述：需要一个“MI-shaped future-aware reward-based CPT”

综合上述工作和我们之前的讨论：

- NTP/CE 在结构上是 **短视 + 稀疏 0/1 奖励**（只奖励 gt token）；
- MTP/FSP 证明了在预训练阶段引入“多步未来”或“未来摘要”有助于长程规划
- Thinking Tokens 表明 **互信息峰值 token 对推理至关重要**，MI 是一个良好的监控甚至 shaping 信号；
- RLVR 在 post-training 阶段主要是 **reweight base model 的推理路径**，很难凭空创造新模式；

- AvataRL 和 RLP 证明了在 CPT 阶段引入 RL 风格的 reward 是可行的，但前者主要偏向“监督+critic 蒸馏”，后者依然围绕 CE+CoT rollout。

因此我们的目标是：

> 在 Continuous Pre-Training 阶段，在保持 teacher forcing 高效并行的前提下，引入面向未来的互信息稠密奖励信号，将“未来摘要世界模型”蒸馏进 LM 表征，并对 token 分布做 future-aware 的 reward shaping，从而提升模型的长程规划和推理能力。

2. 方法：Future-aware MI Shaping under Teacher Forcing

2.1 基本设定与记号

- 训练语料：文本序列 $x = (x_1, \ldots, x_T)$，来自 CPT 数据（code+math+general corpora）；
- 状态：

$$s_t = x \leq t, \quad t = 1, \ldots, T$$

- 主 LM 的隐藏与策略分布：

$$h_t = h_\theta(s_t), \qquad \pi_\theta(a \mid s_t) = \mathrm{softmax}(W h_t)_a$$

- Future encoder 给出的未来摘要：

$$F_t = F_\phi(x_{t+1:t+L}) \ \text{ or } \ F_\phi(x_{t+1:T})$$

整体目标：在 teacher forcing 下，同时优化

1. NTP/MTP loss（保持基本 LM 能力）；
2. CPC-style 互信息目标（让 h_t 含有 future 信息）；
3. token-level future-aware reward（在 top-k token 之间做 credit assignment）。

为什么选用互信息（Mutual Information，MI）？ 在FSP 中的一段话：

> Why Future Summary Prediction? The key advantage of future summary prediction is its ability to reduce dependence on teacher forcing when modeling long future sequences. **To intuitively measure teacher forcing, consider for each ground-truth token exposed to the model, how much information is the model required to predict about unseen tokens? If the model predicts more such information, then we have reduced teacher forcing.** Next-token prediction (NTP) uses the highest degree of teacher forcing, since the model always conditions on ground-truth histories to predict just the next token. Multi-token prediction (MTP) partially relaxes this by asking the model to predict short blocks of future tokens, thereby reducing teacher forcing locally. However, MTP remains constrained by the short horizon of its predictions. In contrast, our proposed approach predicts summaries of long future sequences, substantially reducing teacher forcing by requiring the model to reason about rich, global properties of the target trajectory.

紫色的话正好是和互信息的想法一致。

X，Y互信息 $I(X;Y)$的含义——在已知 Y 的情况下，X 的不确定性（信息量）减少了多少。

在我们的场景中，假设我们在时间步 t，我们得到隐藏层表征 h_t。我们需要预测未来目标 F_t。那么“每暴露一个真值 token，模型需要预测未见过的 token 的信息”可以形式化为：

$$I(h_t; F_t) = H(F_t) - H(F_t|h_t)$$

我们已经有了当前隐藏表征 h_t，如果要预测未来 tokens，还需要多少信息量。现在我们把未来 tokens 包含的信息用 F_t 表示。 $H(F_t)$就是未来 token 的信息量。

- NTP：$T_t = x_{t+1}$（下一个 token）
- MTP：$T_t = (x_{t+1}, \ldots, x_{t+k})$（短块）
- FSP：$T_t = S_t = F_\phi(x_{t+2:T})$

这个量 **越大**，说明在当前步，模型必须从前缀里**推断更多未来结构**（而不是等到后面“喂真值”再学），因此**对 teacher forcing 的依赖越小**。

2.2 Future encoder 的选择：在 CPT 场景下的几种方案

我们在 FSP 讨论里吐槽过 ReLM，本节把几种候选 future encoder 列出来并分析优缺点。

▼ 2.2.1 方案 A：FSP 原版的 Reverse LM（ReLM）

FSP 的“learned summary”方案：训练一个**从右到左的 LM**，将 future 片段 $x_{t+1:T}$，输入 RevLM，取最后一层 hidden 的某种池化作为 F_t。

$$\mathcal{L}_{\text{RevLM}}(x, Q_\psi) = -\mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T-1}\log Q_\psi(x_{t+1}|x_{\geq t+2})\right]$$

$$F_t = g_h(g_s(x_{\geq x_{t+2}})$$

LM训练辅助损失：

$$\parallel A_\phi(x_{\leq t}) - g_h(g_s(x_{\geq x_{t+2}}))\parallel_2^2$$

其中 g_s, g_h 是 Q_ψ的头，用于提取表征。

- 优点：
  - 未来摘要是真正“只看未来”的模型给出的，因果结构干净；
  - 已经在 FSP 实验中证明对长程生成有效。
- 缺点：
  - 需要额外训练一个大 ReLM（右到左），工程开销大；
  - FSP 原文把 LM hidden 与 ReLM hidden 通过简单回归拉近，存在“表征空间不匹配”的风险；
  - 从直觉上，右到左的 inductive bias 跟我们最终的前向 LM 有点不自然。

这个方案可以作为 **理论对照 / sanity baseline**，但不一定是最佳工业方案。

▼ 2.2.2 方案 B：Prefix-Query Pooler + 冻结 forward LM 隐藏

参考 FSP 中的另一种实现思路：用现有 LM 的 hidden 来构造未来摘要，而不是训练 ReLM。

简单版本：

- 用同一个 LM（forward）在 teacher forcing 下跑完全句，得到所有位置的 hidden：

$$h_i^{\text{full}} = h_\theta(x_{\le i}), \quad i = 1, \dots, T$$

- 对于每个 t，把 future 片段的 hidden 喂给一个 prefix-query pooler，例如：

$$F_t = \text{Pooler}_\psi\left(h_{t+1}^{\text{full}}, \dots, h_{t+L}^{\text{full}}\right)$$

Pooler 可以是：

- 简单的 mean/max pooling；
- 一个以“prefix embedding”为 query 的 cross-attention；
- 一个小 Transformer over future tokens。
- 训练方式：
  - 一种做法是 **冻结 LM，单独训练 Pooler**；
  - 更强一点的做法是 LM+Pooler 端到端 co-train，但要小心训练稳定性。

优点：

- 不需要额外 ReLM，完全复用 forward LM 的表示；
- Future summary 在几何空间上自然与 LM 某一层隐藏对齐，更适配后续 CPC；
- 结构简单，易于在 CPT pipeline 中加入。

缺点：

- 需要对整句 forward 一遍才能得到 future hidden，训练时 compute 增加；
- 如果 LM 不冻结，Pooler 和 LM 纠缠得比较紧，有可能造成 optimization coupling（需要 careful schedule）。

这是一个非常自然的 CPT-friendly 方案，建议作为主 method 的 default future encoder

2.3 表征层：CPC-style 互信息目标 $\mathcal{L}_{\text{CPC}}$

借鉴 ***Representation Learning with Contrastive Predictive Coding*** 与 Thinking Tokens 的做法，我们在 **state 表征层**明确加入一个 MI 下界目标：

- 取 LM 的某一层 hidden 作为 h_t，future encoder 给出 F_t；
- 定义打分函数 $f_\omega(h_t, F)$（dot product 或小 MLP）；

> 也有一些论文参考：
> Sequence-level Large Language Model Training with Contrastive Preference Optimization
> Self-Supervised Alignment with Mutual Information
> Learning Transferable Visual Models From Natural Language Supervision

- 用 in-batch negatives 构造 InfoNCE loss

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\left[\log \frac{\exp(f_\omega(h_t, F_t))}{\sum_{(i,t')\in\mathcal{N}\cup\{(i,t)\}} \exp f_\omega(h_t, F_t^{(i)})}\right]$$

其中 $\mathcal{N}$ 是负样本集合（例如 batch 内其他位置的 future summary）。

负例的构造方式：每个样本中随机 mask 一些

考虑其他估计 MI 的方式?

解释：

- $\mathcal{L}_{\text{CPC}}$ 不直接更新 logits，只是强迫 h_t **携带足够多关于其未来摘要 F_t 的信息**；
- 从 Thinking Tokens 的视角看，这相当于在 CPT 阶段提前塑造“MI peaks-friendly”的表征结构。

在训练过程中，有关 MI 的估计以及指标监控也有一定的注意事项，可以参考：

> On Variational Bounds of Mutual Information

2.4 动作层：MI-shaped token-level reward $\mathcal{L}_{\text{token-RL}}$

接下来是我们这篇工作最核心的一块：在 **不破坏 teacher forcing 并行** 的前提下，对 top-k token 加一个 future-aware 的 reward。

1. 在每个位置 t，从当前 policy 分布取 top-k：

$$A_t = \text{TopK}_k\big(\pi_\theta(\cdot \mid s_t)\big) \cup \{x_{t+1}\}$$

2. 定义一个“虚拟 rollout head”，输入为 $(h_t, e(a), F_t)$（分别是当前 hidden、候选 token embedding、future summary），输出一个 scalar score：

$$s_t(a) = f_\psi(h_t, e(a), F_t)$$

例如：

$$f_\psi(h, e, F) = v^\top \sigma(W_h h + W_e e + W_F F)$$

或者：

$$\hat{F}_t(a) = W_2 \sigma(W_1 [h_t; e(a)])$$

将 h_t 和当前 token a 的表征拼接起来，经过 MLP / Transformer block。 参考DeepSeek MTP

$$s_t(a) = \cos(\hat{F}_t(a), F_t)$$

所以在语义上， $\hat{F}_t(a)$ 的含义：

> “在当前 state hth_tht，如果我采取动作 a，我预测到的未来摘要是什么样”。

一旦你把上面的监督加上，就可以把 $\mathrm{sim}(\hat{F}_t(a), F_t)$ 解释成：

> “如果我选 a，预测的未来 和 真实/理想的未来 有多接近？”

这和 RL / Q-learning 里干的事是一模一样的：

- Q-learning 里 $Q_\theta(s, a) \approx \mathbb{E}[G | s_t = s, a_t = a]$ 是“回报预测”；
- $\hat{F}_t(a) = g_\psi(h_t, e(a))$ 是“未来摘要预测”。**类似于优势估计？**

3. 把 score softmax 成 reward 分布：

$$r_t(a) \propto \mathrm{softmax}(s_t(a)/\tau), \quad a \in A_t$$

就是在说：

> “奖励那些 不会把未来带偏太多 的 token（ $\hat{F}_t(a)$ 接近 ），
> 惩罚那些会把未来带偏的 token（ $\hat{F}_t(a)$ 离 F_t 很远）。”

注意这里 F_t 被看成一种 **“理想未来 / teacher 的 future”**，而不必严格理解为“在 action=a 下真实发生的 future”。这是一个 **teacher-forcing-friendly的近似**：

- 真正的 counterfactual future $F_t(a)$ 我们没法算（那就要 rollout+FSP）；
- 但我们假设：合理的 token（比如同义表达、小改写）不会太改变 high-level future summary，于是用 F_t 当 “理想目标”。

4. 最终的 token-level reward loss:

$$\mathcal{L}_{\text{token-RL}} = -\mathbb{E}_{x,t}\left[\sum_{a\in A_t} r_t(a)\log \pi_\theta(a \mid s_t)\right]$$

几点说明：

- 对于 ground-truth token $a = x_t$ ，我们可以在 $s_t(a)$ 中加一个 small bias，例如：

  $$s'_t(a) = s_t(a) + \lambda g_t \cdot 1\{a = x_{t+1}\}$$

  避免完全忽略监督信号；
- 由于 reward 是 dense 的，**top-k 中的非 gt token 也能根据其对未来的兼容性获得正向信号**；
- 训练流程仍然是单次 forward（没有显式 rollout），所有 (t, a) 都可以并行计算 $s_t(a)$。

从 RL 视角：这相当于在 action space 上学习一个 “future-compatible Q-like scorer”，然后用 reward-weighted log prob 优化 policy。与 AvataRL 的差异在于：我们的 reward 不依赖外部 critic，而是由 **future encoder + MI 结构** 内生出来的。

2.5 总体目标与训练流程

总 loss 可以写成：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NP/MTP}} + \lambda_{\text{InfoNCE}}\,\mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{RL}}\,\mathcal{L}_{\text{token-RL}}$$

- $\mathcal{L}_{\text{NTP/MTP}}$：标准 CE +（可选）MTP head，用于基本语言建模与推理 token 的监督；

- $\mathcal{L}_{\text{InfoNCE}}$：显式互信息目标，训练 future-aware 世界模型；
- $\mathcal{L}_{\text{token-RL}}$：future-aware reward shaping，训练 token-level 策略。

训练策略可以有两种：有待商榷

1. **两阶段：world-model → policy**
   - Stage 1：训练 LM + future encoder + CPC（$\mathcal{L}_{\text{NTP/MTP}} + \lambda_{\text{CPC}}\mathcal{L}_{\text{CPC}}$），得到一个 future-aware base model；
   - Stage 2：冻结 future encoder，只加上 $\mathcal{L}_{\text{token-RL}}$ 对 policy 进行 reward shaping（$\mathcal{L}_{\text{NTP/MTP}} + \lambda_{\text{RL}}\mathcal{L}_{\text{token-RL}}$）。
2. **单阶段联合训练**：
   - 从一开始就 jointly optimize 三个项，只是逐步调大 λ_{RL}、降低 λ_{InfoNCE}。

建议主实验采用**两阶段设置**，更便于分析“world model vs policy”的贡献。

3. 实验设计

3.1 核心科学问题 & 假设

RQ1. 在相同 CPT 数据和 compute 下，我们的 MI-shaped future-aware CPT 能否在**长程规划/推理任务**上优于 NTP/MTP/FSP 基线?

RQ2. 在 NTP failure mode 的 toy task 上（如 Pitfalls-of-NTP 的 planning 任务），我们的方法能否像 MTP 一样修复结构性问题，同时保持鲁棒推理?

RQ3. 从互信息视角（Thinking Tokens 风格分析），我们的 CPT 是否改变了“MI 峰值 token”的分布，使得更多规划相关 token 成为“thinking tokens”?

RQ4. 我们的方法与 RLP/AvatarRL 这类 RL-pretraining 框架相比，在效果、计算成本与适用性上各有什么差异?

3.2 模型与数据

- 模型规模：
  - 首轮在 300M–1B 级别的 decoder-only 模型上验证（例如 Qwen1.5 0.5B / GPT2-style 自训模型）；
  - 如效果显著，可在 3B–7B 级别模型上进一步验证。

- 数据分布：
  - CPT stage：通用语料 + math/coding-heavy 子集（code, math word problems, formal proofs 等），类似 RLP / FSP 设置
- 训练设置：
  - baseline 与我们的方法使用**完全相同的数据配比与 token budget**；
  - 记录训练过程中 NLL、MI 曲线等指标。

3.3 Toy Problem：针对 NTP 短视的合成任务

参考 ***Pitfalls of NTP*** 与 ***Going beyond the creative limits of NTP***，设计几类 toy 任务：

1. **Path-Star / 图规划任务**
   - 类似 Pitfalls-of-NTP 的 minimal planning task：输入是一条图上的路径描述或邻接表，输出是正确的路径 / 终点；
   - 教师 forcing 条件下，标准 NTP 会倾向于“贪心复现局部边”，而无法学会 global 规划；
   - 指标：teacher-forcing 精度、autoregressive 精度、对扰动/反向图的鲁棒性。
2. **Sliding-window / global constraint 任务**
   - 需要记住一个很早出现的信息，并在很晚的 token 上做决策（典型长程 credit assignment）
   - 类似 “最后一个 token 是否等于前 N 个 token 的某种函数” 的任务。
3. **Algorithmic creativity task 的简化版**（受 ICML 2025 creative-limits 影响）
   - 抽象知识图类任务：要求模型在已有 graph 上找一条新路径连接两个概念；
   - Pattern construction：要求模型设计一个符合某种全局约束的字符串模式。

在这些任务上，我们希望看到：

- NTP baseline 复现 Pitfalls 里的 failure mode；
- MTP/FSP 有一定修复；
- 我们的 MI-shaped CPT 能进一步提升性能，尤其是在“规划正确率 vs. 生成多样性”上。

3.4 真实任务：推理、规划与长上下文

1. **数学推理**：GSM8K、SVAMP、MATH-mini、AMPS subset 等；
2. **代码推理**：HumanEval、MBPP、CodeContests subset 等；
3. **长上下文任务**：长文问答、needle-in-haystack、长代码补全；
4. **结构性规划任务**：表格推理、多跳问答等。

评测策略：

- 标准 accuracy / pass@k（包括大k，对齐 RLVR 的分析方式）；
- 推理路径分析：CoT/自然语言 reasoning 的质量（可以用 small verifier 评估 logical consistency）；
- 生成多样性与稳定性（特别是 creative 任务）。

3.5 对比基线与 ablation

主 Baselines

1. **NTP-only CPT**（标准 CE 预训练）；
2. **NTP + MTP**（例如 DeepSeek-V3 风格的多头未来 token 预测）
3. **NTP + FSP**（带 handcrafted 和 learned summary 两种版本
4. **Reward-based CPT w/o future**：AvataRL-style 的 reward weighting，但 reward 只由 reality expert + critic 提供，不用 future summary。
5. ~~（如果算力允许）**RLP-style CPT**：参考 NVIDIA RLP，在 CPT 阶段 reward chain-of-thought。~~

Ablation 变量

1. **MI vs RL**
   - $\mathcal{L}_{\text{CPC}} + \mathcal{L}_{\text{NTP/MTP}}$（只有 representation-level MI，没有 token-level reward）；
   - 仅 $\mathcal{L}_{\text{token-RL}} + \mathcal{L}_{\text{NTP/MTP}}$（只有 reward-shaping，没有 CPC）；
   - 两者都加（完整方法）。
2. **Future encoder 选择**
   - 方案 B（forward LM hidden + prefix-query pooler，默认）；
   - 如算力允许，方案 A（ReLM）做 FSP-style baseline。
3. **Future horizon L**

- 不同 L（例如 32/64/128/256）对长程任务/短程任务的影响。

4. **Top-k 规模与温度 τ**
   - 比较 small k（如 4/8） vs large k（如 32）对 reward landscape 平滑程度与训练稳定性的影响；
   - 不同 τ 如何影响“探索 vs 利用”。
5. **两阶段 vs 联合训练**
   - Stage1: CPC-only → Stage2: token-RL；
   - vs直接 joint training。

3.6 分析工具：MI 曲线与 Thinking Tokens

借鉴 Thinking Tokens 的分析框架，我们可以在训练前后，用 MI 作为观测：

- 估计 $I(h_t, y)$ 随生成步 t 的变化曲线（使用作者公开代码/近似方法）；
- 分析 MI peaks 对应的 token 类型：是明确的思考词 / 分隔符 / 结构性符号（如“if/for/return”、“因此”、“→”）；
- 比较 NTP baseline、FSP-only、我们方法在 MI 峰值分布上的差异（峰值数量、位置、与关键 reasoning step 的对齐程度）。

这部分可以作为“分析章节”，也可以是另一个小的 side paper。

4. 预期结果与分析框架

4.1 如果实验成功，我们希望看到什么？

1. **Toy tasks 上**：
   - 在 Pitfalls-of-NTP 的 minimal planning 任务上，NTP 出现 failure mode（如文中所示），MTP/FSP 有改善，而我们的 MI-shaped CPT 进一步提升（尤其在 generalization / robustness 上）。
   - 在 creative task 上（受 creative-limits 启发），我们的模型能在保持 coherence 的前提下，生成更多样、更“远”的 pattern。
2. **真实推理任务上**：
   - 在 math/code benchmark 上，pass@1 有提升，同时在大 kkk 的 pass@k 下，**base+MI CPT 的能力边界有所拓展**，而不仅仅是 reweight（对 RLVR 的结论形成对比/补充）。

- 在长上下文任务上，长程依赖的准确率提升，错误更少集中在“早期信息丢失”与“planning failure”上。

3. **MI/Thinking Tokens 分析**：
   - MI 峰值在我们模型中更集中于结构性/规划性 tokens，而不只是显式 CoT 关键词；
   - 在长程任务中，MI peaks 更早出现，或在关键决策点前后集中，说明模型在更早阶段已经“考虑了未来”。

4.2 如果发现“效果有限”，我们怎么解释？

若与 MTP/FSP 相比，我们的方法在主 benchmark 上提升有限，可以做以下分析：

- 从 RLVR 角度：可能我们的 MI-shaped reward 仍然更多是在 reweight base pattern，而没有构造新的推理模式；
- FSP 本身已经在较大程度上让 hidden 携带未来信息，我们的额外 MI/RL 只是在边际上改善 reward landscape；
- Future encoder 的质量/设计可能是瓶颈（例如 F_t 太粗糙、MI 估计噪声太大）；
- MI measurment（InfoNCE）本身是 noisy 下界，不能过于把“互信息数值”当真，应更多看 representation 结构和行为层面的变化。

这时可以回收价值：

- 即便主 metric 提升有限，我们依然可以写一篇偏 **分析/负结果** 的 paper：
  - “MI-shaped reward-based CPT 在多大程度上改变了表征与推理轨迹？”
  - 与 RLVR“RL 不引入新推理”的结论形成互文。

4.3 与 RLP / AvataRL 的关系

- RLP：CoT 作为 action，用信息增益 reward 奖励“有用思考”；
- AvataRL：top-k + critic + reality expert，reward-weighted log-prob；

我们的区别：

1. **阶段**：我们重点在 CPT，而非 post-training RLHF（与 RLVR 区分）；
2. **信号源**：reward 不来自 external critic 或 problem verifier，而是来自 **未来摘要世界模型 + MI 结构**；
3. **形式**：我们没有显式 CoT rollout，仍然保留 teacher forcing 的高并行度，适合大规模 CPT；

4. **目标**：我们关注的是 **representation-level 的 future-aware bias + action-level reward shaping**，希望能真正拓展 reasoning capability boundary，而不是纯 reweight。

4.4 未来工作

如果这条线跑通，还有很多自然扩展：

- 多 horizon future summaries（近 horizon vs 远 horizon），对应多阶段 planning；
- 把 MI-shaped reward 与 RLP 的 CoT reward 结合：既 reward 思考过程也 reward 对未来的兼容性；
- 在 code/formal verification 场景下，用形式语义/证明状态替代 F_t，把逻辑世界模型塞进 LM；
- 结合 interpretability：分析不同维度 reward 下形成的 circuits，与你“Neural Circuits for Alignment”项目打通。

实验一

对比 ce

和我们的方法

数据 open-thoughts

baseline：CE-CPT

ours：