

- 作者：GaoZheng
- 日期：2025-09-22
- 版本：v1.0.0

摘要

本文面向字符级 POMDP 场景，系统化整理离散动作 SAC 的实现细节：策略/价值网络结构、温度/熵目标的自适应、Top-p 采样与合规 Mask 的协同，以及 CQL/BC/DAGger/EMA 等稳定训练技巧。结合生产日志与指标，给出从冷启动到稳态的调参与收敛路径，并讨论长序列与约束采样下的可观测性折中。

下面给出**独立、可落地、可审计**的改进版规范（V2）。已内嵌所有纠偏：**温度自适应符号修正**、**(1-done) 截断**、**奖励尺度化**、**EMA-NCE 防漂移**、**Top-p 近似**、**数值稳定**、**CQL 正则**、**教师冲突治理**。文稿可直接做为实现与灰度上线的执行依据。

定位：面向生产的**字符级 POMDP**与**离散最大熵 SAC**训练方案。内置**硬掩码合规**、**期望式备份 + Top-p 近似**、**温度自适应（修正）**、**双缓冲 + BC + DAGger**、**InfoNCE 奖励（EMA 目标编码器）**、**在线 KPI/SLI**、**CQL 正则抗 OOD**与**PopArt 奖励尺度化**。

目标：在**可控合规**前提下，提升**段落级一致性**与**域外泛化**，实现**灰度可观测**与**回滚可控**。

- 温度自适应修正**：采用 $\log \alpha \leftarrow \log \alpha + \eta_{\alpha} (H_{\text{tgt}} - H(\pi))$ ；当熵不足时提升 α 。
- 终止截断**：目标值加入 $(1 - \text{done})$ ；终止步不 bootstrap。
- Top-K \rightarrow Top-p**：按覆盖率 p 近似期望（默认 $p = 0.98$ ）；选集 `detach()`，仅用于值目标，策略项可选全量或同一 Top-p。
- 奖励尺度化**：对覆盖与 NCE 奖励启用 **PopArt/EMA 标准化**，抑制尺度漂移。
- InfoNCE 防漂移**：使用 **EMA 目标编码器** \bar{f} 与负样本队列。
- 数值稳定**：稳定化 softmax、 ϵ -clip、掩码 `-1e9`、logits 平移。
- CQL 正则（可拨码）**：缓解 demo/agent 混分布下的 OOD-Q 过估。
- 教师并轨治理**：教师动作与掩码冲突**直接拒收或重标**，DAGger 指标纳管。

1. 符号与范围

- 字表 Σ （含特殊符号），合法动作集 $\mathcal{A}(s) \subseteq \Sigma$ 。
- 折扣 $\gamma \in (0, 1)$ ；温度 $\alpha = e^{\tilde{\alpha}}$ ， $\tilde{\alpha} = \log \alpha$ 。

- 策略 $\pi_\theta(a \mid o)$; Twin-Q: Q_{ϕ_1}, Q_{ϕ_2} ; 目标网络 Q'_{ϕ_1}, Q'_{ϕ_2} 。
- 回放: $\mathcal{D}_{agent}, \mathcal{D}_{demo}$; 混采比例 $\rho \in (0, 1)$ 。
- 合法集熵目标: $H_{tgt}(s) = \kappa \cdot \log |\mathcal{A}(s)|$ (nats) , $\kappa \in [0.7, 1.2]$ 。

2. 任务建模 (无泄漏 POMDP)

POMDP 七元组 $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$ 。

- **状态** $s_t = (h_{t-1}, u_{t-1})$:
 $h_{t-1} \in \Sigma^m$ 为最近 m 字滑窗; $u_{t-1} \in \mathbb{R}^{d_c}$ 为上下文摘要 (默认 GRU) 。
- **动作** $\mathcal{A} = \Sigma$: 单字符。
- **观测**:

$$o_t = \mathcal{O}(s_t) = \text{Enc}(h_{t-1}) \oplus u_{t-1} \quad (\text{无参考对/未来泄漏})$$

- **转移** (确定性) :

$$h_t = \text{shift_append}(h_{t-1}, a_t), \quad u_t = \text{GRU}(u_{t-1}, \text{Emb}(a_t))$$

若触发**教师覆盖** (整段 g_t) :

$$(h_t, u_t) \leftarrow (g_t, \text{GRU_seq}(u_{t-1}, \text{Emb}(g_t)))$$

- **终止**: 达步长 T 或触发非法硬约束早停。
- **奖励**: 见 §5。

3. 合规与动作空间 (稳定硬掩码)

- 合规函数 $\text{Compliance}(h_{t-1}, a) \in \{0, 1\}$ 。
- 合法集 $\mathcal{A}(s_t) = \{a \in \Sigma \mid \text{Compliance} = 1\} \cup \{\langle eos \rangle\}$ 。
- **稳定掩码到 logits**:
 - 先做平移: $\hat{z} = z - \max(z)$;
 - 掩码: $\tilde{z}_i = \hat{z}_i$ 若 $i \in \mathcal{A}(s_t)$, 否则 -10^9 ;
 - softmax: $\pi = \text{softmax}(\tilde{z})$; 概率做 ϵ -clip ($\epsilon = 10^{-8}$) , 避免 $\log 0$ 。
- 合规器“过严回退”: 当 $|\mathcal{A}(s)| < 3$, 切换**分层白名单** (字符类→细粒度字表) + $\langle eos \rangle$ 。

4. 模型结构

- **Encoder**: 字符嵌入 $E: \Sigma \rightarrow \mathbb{R}^d$; GRU/Conv 得上下文向量 $c_t \in \mathbb{R}^d$ 。
- **策略头**: $z = \text{MLP}_\theta(c_t) \in \mathbb{R}^{|\Sigma|} \rightarrow \text{掩码} \rightarrow \text{softmax}$ 。
- **Twin-Q**: $Q_{\phi_i}(o, a) = \text{MLP}_{\phi_i}([c_t; \text{Emb_act}(a)])$ (动作嵌入替代 one-hot)。
- **目标网络**: ϕ_i 指数滑动更新 (EMA)。

5. 奖励设计 (尺度化与防漂移)

设参考文本的 n -gram 序列为 $\{g_t^{(n)}\}$ 。

(1) n -gram 覆盖 (滑窗 W)

$$\text{cov}_t = \frac{1}{|S_t|} \sum_{x \in S_t} \mathbf{1}[x \in \mathcal{G}_t], \quad S_t = \text{Agent } n\text{-gram}(t, W), \quad \mathcal{G}_t = \text{Ref } n\text{-gram}(t, W)$$

(2) 对比相似 (InfoNCE, EMA 目标编码器)

正样 $x^+ = g_t^{(n)}$, 难负样 $x^- \in \mathcal{N}_t$ 来自**队列缓冲**; 表示用在线编码器 f 与**目标编码器** \bar{f} (EMA: $\bar{f} \leftarrow m\bar{f} + (1-m)f$)。

$$\text{ncc}_t = \log \frac{\exp(\langle f(S_t), \bar{f}(x^+) \rangle / \tau)}{\exp(\langle f(S_t), \bar{f}(x^+) \rangle / \tau) + \sum_{x^- \in \mathcal{N}_t} \exp(\langle f(S_t), \bar{f}(x^-) \rangle / \tau)}$$

(3) 字符二元奖励 (拓扑记忆)

对字符模式, 构造上一字符与目标字符形成的二元组 $b_t = s_{t-1}^{(1)} c_t$ 。若 b_t 命中

`data/chinese_frequency_word.json` 或 `data/chinese_name_frequency_word.json` 提取的二字词集合 \mathcal{L} , 则给予额外奖励

$$\text{bonus}_t = \lambda_{\text{bigram}} \cdot \mathbf{1}[b_t \in \mathcal{L}], \quad \lambda_{\text{bigram}} = 1.0,$$

其中 \mathcal{L} 为上述两个词表的并集 (过滤为二字词), 必要时可合并原文滑窗补充样本。二元组 b_t 由上一目标字符与当前动作字符拼接而成, 该奖励直接累加到字符模式的 `soft` 组件, 促使策略优先记忆原文的非交换邻接字符组合。同时, 将质量信号按照 0.5/0.25 的权重注入基础与潜在分数, 使高质量字符动作在硬指标上得到体现。

(4) 洁净/非法罚

$$\text{ill}_t = \mathbf{1}[a_t \notin \mathcal{A}(s_t)], \quad \text{garble}_t = \text{Garble}(a_t)$$

(5) 奖励尺度化 (PopArt/EMA 标准化)

对 $\text{cov}_t, \text{ncc}_t$ 应用

$$\mathcal{N}(x_t) = \frac{x_t - \mu_t}{\sigma_t + \epsilon}, \quad \mu_t = (1 - \beta)\mu_{t-1} + \beta x_t, \quad \sigma_t^2 = (1 - \beta)\sigma_{t-1}^2 + \beta(x_t - \mu_t)^2$$

$\beta \in [10^{-4}, 10^{-2}], \epsilon = 10^{-8}$ 。

(6) 步级奖励 (无恒零项)

$$r_t = \lambda_{\text{cov}} \cdot \mathcal{N}(\text{cov}_t) + \lambda_{\text{ncc}} \cdot \mathcal{N}(\text{ncc}_t) - \lambda_{\text{gar}} \cdot \text{garble}_t - \lambda_{\text{ill}} \cdot \text{ill}_t$$

非法动作可选 `done=True` (硬边界), $r_t = -\lambda_{\text{ill}}$ 。

建议: $n \in \{3, 4\}, W = 64, \tau = 0.07$ 。

6. 离散最大熵 SAC (期望备份 · Top-p)

软值函数 (目标网络 + 当前策略)

$$V_{\text{soft}}(s') = \sum_{a' \in \mathcal{A}(s')} \pi_{\theta}(a'|o') \left[\min_i Q'_{\bar{\phi}_i}(o', a') - \alpha \log \pi_{\theta}(a'|o') \right]$$

Top-p 近似 (降耗·覆盖率驱动)

取最小集合 $\mathcal{P}(s') \subseteq \mathcal{A}(s')$ 使 $\sum_{a' \in \mathcal{P}} \pi(a'|o') \geq p$; 定义 $\pi_p \propto \pi \cdot \mathbf{1}[a \in \mathcal{P}]$ (重归一化)。选集 `detach()`。

$$\hat{V}_{\text{soft}}(s') = \sum_{a' \in \mathcal{P}(s')} \pi_p(a'|o') \left[\min_i Q'_{\bar{\phi}_i}(o', a') - \alpha \log \pi_p(a'|o') \right]$$

Bellman 目标与损失 (Huber 推荐)

$$y_t = r_t + \gamma(1 - \text{done}_t) \hat{V}_{\text{soft}}(s_{t+1}), \quad \mathcal{L}_Q = \mathbb{E} \sum_{i=1}^2 \text{Huber}(Q_{\phi_i}(o, a) - y_t)$$

策略目标 (期望式)

选项 A (全量合法集, 精确):

$$\mathcal{L}_{\pi} = \mathbb{E}_o \sum_{a \in \mathcal{A}(s)} \pi_{\theta}(a|o) \left[\alpha \log \pi_{\theta}(a|o) - \min_i Q_{\phi_i}(o, a) \right]$$

选项 B (与值侧一致的 Top-p 近似, 覆盖率 $p \geq 0.98$): 将求和域换为 $\mathcal{P}(s)$ 并重归一化为 π_p 。

温度自适应 (修正版·期望式)

$$\tilde{\alpha} \leftarrow \tilde{\alpha} + \eta_{\alpha} \cdot \mathbb{E}_o [H_{\text{tgt}}(s) - H(\pi(\cdot|o))], \quad H(\pi) = - \sum_a \pi \log \pi$$

保证熵不足 $\rightarrow \alpha$ 上升, 方向正确。并在线监控 $\alpha \in [10^{-4}, 2]$ 。

CQL 正则 (可选拨码)

缓解 OOD-Q:

$$\mathcal{L}_{\text{CQL}} = \lambda_{\text{CQL}} \cdot \left(\mathbb{E}_o \left[\log \sum_a e^{Q_{\phi}(o,a)} \right] - \mathbb{E}_{(o,a) \sim \mathcal{B}} [Q_{\phi}(o,a)] \right)$$

总 critic 损失: $\mathcal{L}_Q^{\text{tot}} = \mathcal{L}_Q + \mathcal{L}_{\text{CQL}}$ 。

目标网络软更新

$$\bar{\phi}_i \leftarrow \tau \phi_i + (1 - \tau) \bar{\phi}_i$$

7. 教师并轨 (双缓冲 + BC + DAgger)

- **双缓冲**: 教师样本入 $\mathcal{D}_{\text{demo}}$ (`is_demo=1`) , 代理样本入 $\mathcal{D}_{\text{agent}}$ 。
- **混采**: 分层/配额采样, 实际批 $\mathcal{B} = \mathcal{B}_{\text{agent}} \cup \mathcal{B}_{\text{demo}}$, 比例 $\rho : (1 - \rho)$ 。
- **BC 辅助** (仅 demo) :

$$\mathcal{L}_{\text{BC}} = \lambda_{\text{BC}} \cdot \mathbb{E}_{(o,a^*) \in \mathcal{B}_{\text{demo}}} [-\log \pi_{\theta}(a^*|o)]$$

- **策略总损失**: $\mathcal{L}_{\pi}^{\text{tot}} = \mathcal{L}_{\pi} + \mathcal{L}_{\text{BC}}$ 。
- **DAgger 调度**: `teacher_ratio` 线性退火 (1.0 \rightarrow 0.1) 。
- **冲突治理**: 若教师动作与掩码冲突, 样本拒收或映射到最近合法替代并标注 `is_relabeled=1` (分桶监控) 。

可选替代: IQL/AWAC 优势加权减少显式 BC 扭曲 (拨码试验, 不纳入最小必需集) 。

8. 最小可用训练循环（伪代码·V2）

```
# === 初始化 ===
init_policy(theta); init_q(phi1, phi2); init_target_q(bar_phi1 <- phi1, bar_phi2 <- phi2)
log_alpha = init_log_alpha(); alpha = exp(log_alpha)
replay_agent, replay_demo = RB(), RB()
popart_cov, popart_nce = PopArt(), PopArt()    # 维护 (mu, sigma)
neg_queue = QueueK()                          # NCE 难负队列
ema_target_f = init_target_encoder(f)         # EMA 目标编码器

for episode in range(E):
    s = env.reset()
    for t in range(T):
        # --- 前向与掩码（数值稳定） ---
        logits = policy_logits(theta, o(s))
        logits = logits - logits.max()          # 平移
        logits = masked_fill_illegal(logits, s, -1e9)    # 硬掩码
        pi = softmax(logits).clamp(min=1e-8)          #  $\epsilon$ -clip
        a = sample_from(pi)

        s_next, r_raw, done, info = env.step(a)

        # --- 奖励工程 ---
        cov_t = compute_ngram_cov(s, ref, W, n)
        nce_t = compute_infoNCE(f, ema_target_f, S_t, g_t_pos, neg_queue, tau)
        r = (lambda_cov * popart_cov.norm(cov_t)
              + lambda_nce * popart_nce.norm(nce_t)
              - lambda_gar * Garble(a)
              - lambda_ill * is_illegal(a, s))

        # 写缓冲
        if info["is_teacher"]:
            # 教师动作若与掩码冲突：拒收或合法重标（并记录 is_relabeled）
            o_demo, a_demo = o(s), info["teacher_action"]
            if is_illegal(a_demo, s): continue_or_relabel()
            replay_demo.add(o_demo, a_demo, r, o(s_next), done, is_demo=1)
        else:
            replay_agent.add(o(s), a, r, o(s_next), done, is_demo=0)

    # --- 训练 ---
```

```

if ready():
    B_agent = replay_agent.sample(batch_size * rho, stratified=True)
    B_demo = replay_demo.sample(batch_size * (1 - rho), stratified=True)
    B = merge(B_agent, B_demo)

    # Critic 目标 (Top-p 值近似; 选集 detach)
    with no_grad():
        logits_next = policy_logits(theta, o_next(B))
        logits_next = logits_next - logits_next.max(dim=-1, keepdim=True).values
        logits_next = masked_fill_illegal(logits_next, s_next(B), -1e9)
        pi_next = softmax(logits_next).clamp(min=1e-8)

        P_idx = top_p_indices(pi_next, p=top_p).detach()
        pi_p = renorm(pi_next.masked_fill(~P_idx, 0.0))
        q1_t = target_q1(bar_phi1, o_next(B), A_in(P_idx))
        q2_t = target_q2(bar_phi2, o_next(B), A_in(P_idx))
        v_soft = (pi_p * (torch.min(q1_t, q2_t) - alpha * torch.log(pi_p))).sum(-1)

        y = r(B) + gamma * (1 - done(B)) * v_soft

    q1 = q_fn(phi1, o(B), a(B)); q2 = q_fn(phi2, o(B), a(B))
    L_Q = huber(q1 - y) + huber(q2 - y)

    if use_cql:
        L_CQL = lambda_cql * (logsumexp_q(phi1, o(B)) - q1.mean()
                               + logsumexp_q(phi2, o(B)) - q2.mean())
        L_Q = L_Q + L_CQL

    update(phi1, phi2) to minimize L_Q

    # 策略 (可选全量或 Top-p 近似; 保持与值侧一致性)
    logits_curr = policy_logits(theta, o(B))
    logits_curr = logits_curr - logits_curr.max(dim=-1, keepdim=True).values
    logits_curr = masked_fill_illegal(logits_curr, s(B), -1e9)
    pi_curr = softmax(logits_curr).clamp(min=1e-8)

    if policy_use_topp:
        P_idx = top_p_indices(pi_curr, p=top_p).detach()
        pi_eff = renorm(pi_curr.masked_fill(~P_idx, 0.0))
        qmin = torch.min(q_all(phi1, o(B), A_in(P_idx)),

```

```

        q_all(phi2, o(B), A_in(P_idx)))

    else:
        pi_eff = pi_curr
        qmin = torch.min(q_all(phi1, o(B), ALL),
                        q_all(phi2, o(B), ALL))

    L_pi = (pi_eff * (alpha * torch.log(pi_eff) - qmin)).sum(-1).mean()

    # BC (仅 demo)
    L_BC = lambda_BC * cross_entropy_on_demo(pi_curr, a_star(B_demo_only))
    update(theta) to minimize (L_pi + L_BC)

    # 温度 (期望式修正)
    with no_grad():
        H = entropy(pi_curr) # -sum pi log pi
        H_tgt = kappa * log_legal_count(s(B))
        log_alpha += eta_alpha * (H_tgt - H).mean()
        alpha = exp(log_alpha)

    # 目标网络更新
    soft_update(bar_phi1, phi1, tau)
    soft_update(bar_phi2, phi2, tau)

    if done: break
    s = s_next

```

9. 默认参数 (V2 建议)

- **模型**: $d = 128$, GRU hidden = 256。
- **训练**: batch = 2048; $\text{lr}_\pi = 3\text{e-}4$, $\text{lr}_Q = 3\text{e-}4$, $\text{lr}_\alpha = 1\text{e-}4$; $\tau = 0.005$ 。
- **SAC**: $\gamma = 0.995$ (中文长序列折中, 候选: 0.99/0.995/0.997 A/B) ; Top-p = 0.98 (监控覆盖率 $\geq 95\%$) 。
- **熵**: $\kappa = 0.9$; $\alpha \in [10^{-4}, 2]$ 动态。
- **奖励**: $\lambda_{\text{cov}} = 1.0$, $\lambda_{\text{ncc}} = 0.5$, $\lambda_{\text{gar}} = 0.1$, $\lambda_{\text{ill}} = 2.0$; PopArt $\beta = 1\text{e-}3$ 。
- **数据**: $\rho = 0.75$, $\lambda_{BC} = 0.1$, DAgger 线性退火 200k→20k 步。
- **稳定**: 梯度裁剪 0.5; 掩码在 logit 层; nan/inf 钩子开启。
- **CQL**: $\lambda_{\text{CQL}} = 0.5$ (可拨码 0~1) 。

10. 质量 Gate (Go-Live KPI/SLI)

- **稳定性**：三次独立跑无发散；critic Huber 损失稳定下降； α 收敛入 $[10^{-4}, 2]$ 。
 - **泛化**（域外章节）：Top-1/Top-3 字命中、4-gram 覆盖 \geq 基线 +10pp。
 - **合规**：非法字符比 $<0.1\%$ ；**提前终止率** $<1\%$ ；脏尾 = 0。
 - **一致性**：段落一致性分 \geq 无泄漏基线 +8pp。
 - **消融**：去 BC / 去 Top-p / 固定 α 任一项，关键指标下降 $\geq 5pp$ （证实效用）。
- Gate 未达标，一律回滚。

11. 运维与监控（面板字段）

- **策略侧**： $\mathbb{E}[H(s)]$ 、 α 轨迹、Top-p 覆盖率、非法率、提前终止率、 $\max_a \pi(a|o)$ 。
- **价值侧**：critic loss、TD-error 分布、目标/在线 Q 的均值方差、CQL 项。
- **奖励侧**：cov/nce 的均值/方差与 PopArt 统计 (μ, σ) ，漂移告警。
- **合规侧**：掩码命中率、黑名单热更时延、回滚次数。
- **数据侧**：DAgger 比例、 $KL(\pi \parallel \pi_{\text{teacher}})$ 、demo 覆盖率、重标率。
- **质量侧**：BLEU-n、chrF、人工洁净度审计阳性率（分桶）。

12. 兼容性与扩展

- **观测泄漏**：参考对/未来仅用于奖励与评测，不入 o_t 。
- **Actor/Env 一致**：单步分类头；训练与推理语义等价。
- **Top-p 自适应**：当**覆盖率 $<95\%$ **自动提升 p 或转 Top-K（K 自适应）。
- **子词/词级迁移**：动作嵌入保持接口，即插即用。

13. 风险与缓释（生产视角）

风险	影响	概率	缓释
熵不足时 α 方向错误	高	低	已修正更新式 + α 区间监控
终止步 bootstrap 污染	高	低	(1 - done) 截断 + 单测

风险	影响	概率	缓释
InfoNCE 漂移	高	中	EMA 目标编码器 + 队列负样 + 温标固定
Top-p 覆盖不足偏差	中	中	覆盖率 KPI 触发升 p/K ; 选集 <code>detach()</code>
Demo 价值污染	中	中	掩码冲突拒收/重标 + CQL 正则
数值不稳 (nan/inf)	中	中	logits 平移、 ϵ -clip、监控钩子

14. 潜能塑形（不变性声明）

任意有界潜能 $\Phi : \mathcal{S} \rightarrow \mathbb{R}$ 的形状奖励

$$r'_t = r_t + \gamma \Phi(s_{t+1}) - \Phi(s_t)$$

在 $\gamma < 1$ 下不改变最优策略：

$$\arg \max_{\pi} \mathbb{E} \sum_t \gamma^t r'_t = \arg \max_{\pi} \mathbb{E} \sum_t \gamma^t r_t.$$

实现上**限定量级**，避免主奖励被盖过。

备注（实施清单）

- 单测：温度更新方向、(1-done) 截断、Top-p 覆盖率、掩码冲突处理、PopArt 数学一致性。
 - A/B: $\gamma \in \{0.99, 0.995, 0.997\} \times p \in \{0.97, 0.98, 0.99\}$ 。
 - 拨码：`use_cql`、`policy_use_topp`、`ema_target_on`、`illegal_done_on`。
- 执行口径：**先合并本 V2 规范的**必改项**（温度/终止/数稳/奖励尺度/EMA-NCE/Top-p），再做小流量灰度；Gate 不过，自动回滚与快照对比复盘。

15. 实现映射（仓库现状概览）

- 无泄漏观测：**`ArticleEnvironment.reset/step` 在字符模式下返回 `TextObservation(pair[0], "")`，仅暴露上一字符；二元组 `pair=(c_t-1, c_t)` 与目标字符 `c_t` 仅在奖励与日志阶段使用。
- 原地迭代：**日志中的 `prev_summary=c_t-1`、`chapter=c_t`，`raw_action=c_t+1`（若存在）；`source=c_t-1c_t`，符合“非交换临近字符”拓扑。

- **硬掩码数稳：** `TextPolicyNetwork._mask_logits` 将非法 logits 置为 `-1e9`，`first_step_distribution` 提供合法掩码、概率与对数概率输出，直接支撑 Top-p 期望与熵估计。
- **Top-p 期望：** `DemoSACAgent.update` 的 `_select_top_p / _evaluate_q_candidates` 组合在目标和策略两侧均采用截断重归一的概率，保持 $(1 - done)$ 截断和 Twin-Q 最小化。
- **温度自适应：** 维护 `log_alpha` (Adam 优化，学习率可配置)，执行 $\log \alpha \leftarrow \log \alpha + \eta(H_{tgt} - H)$ 并限制 $\alpha \in [10^{-4}, 2]$ ；更新返回实时 `alpha` 供监控。
- **奖励拆分展示：** 日志中 `base/potential/soft` 通过 `_format_reward_component` 自动映射为“满分/负满分/数值”；在字符模式且代理输出与目标对齐时，三项同时显示“满分”。
- **字符二元奖励：** `ArticleEnvironment.step` 将目标字符与当前预测字符拼接成二元组，在字符模式检测其是否存在于 `data/chinese_frequency_word.json` 或 `data/chinese_name_frequency_word.json`，命中时追加 `CHARACTER_LEXICAL_BIGRAM_BONUS=1.0`；若未命中但与教师目标一致，则给予 `0.5` 的回退奖励，并在日志中记录 `lexical_bigram_bonus`。
- **词频补全：** 启动时调用 `_augment_lexical_statistics_with_bigrams` 对词频缓存进行补全，确保原文中出现的二字词至少以频次 1 写回。
- **日志宽度参数：** `character_length_field_width` 控制字符模式日志长度字段，默认 1，可在配置中调节。
- **日志宽度参数：** `character_length_field_width` 控制字符模式日志的长度字段，默认 1，可通过配置调整补零宽度。
- **Trainer 日志同步：** 字符模式下 `DemoTrainer.run` 使用轮次教师对进行日志与教师干预，保证代理观测与回放的一致性。

示例（原文片段“这五个字像一道闪电...”中“意味着什么”的字符展开）：

Step 01 | prev_summary=0001 chars "这"
| chapter=0001 chars "意"
| source=0002 chars "这意"
| action_source=teacher
| raw_action=0001 chars "味"
-> summary=0001 chars "意"
reward=0.803241 (base=+0.000000, potential=+0.000000, soft=+0.803241; 本次获得最高奖励)

Step 02 | prev_summary=0001 chars "意"
| chapter=0001 chars "味"
| source=0002 chars "意味"
| action_source=teacher
| raw_action=0001 chars "着"
-> summary=0001 chars "味"
reward=0.803241 (base=+0.000000, potential=+0.000000, soft=+0.803241; 本次获得最高奖励)

Step 03 | prev_summary=0001 chars "味"
| chapter=0001 chars "着"
| source=0002 chars "味着"
| action_source=teacher
| raw_action=0001 chars "什"
-> summary=0001 chars "着"
reward=0.803241 (base=+0.000000, potential=+0.000000, soft=+0.803241; 本次获得最高奖励)

Step 04 | prev_summary=0001 chars "着"
| chapter=0001 chars "什"
| source=0002 chars "着什"
| action_source=teacher
| raw_action=0001 chars "么"
-> summary=0001 chars "什"
reward=0.803241 (base=+0.000000, potential=+0.000000, soft=+0.803241; 本次获得最高奖励)

Step 05 | prev_summary=0001 chars "什"
| chapter=0001 chars "么"
| source=0002 chars "什么"
| action_source=teacher
| raw_action=0001 chars "? "
-> summary=0001 chars "么"
reward=0.803241 (base=+0.000000, potential=+0.000000, soft=+0.803241; 本次获得最高奖励)

该日志由 DemoTrainer 自动生成，前两行展示观测窗口（历史/目标字符）， raw_action 为策略输出字符， summary 为环境记账后的最新历史，结尾列出奖励拆分，便于人工复核。

许可声明 (License)

Copyright (C) 2025 GaoZheng

本文档采用[知识共享-署名-非商业性使用-禁止演绎 4.0 国际许可协议 \(CC BY-NC-ND 4.0\)](#)进行许可。