

# 字符模式 SAC 的工程实现与数学化描述v2.0.0

- 作者: GaoZheng
- 日期: 2025-09-26
- 版本: v1.0.0

## 摘要

版本 v2.0.0 在 v1 基线之上引入候选采样改进、奖励拆分与度量细化、目标网络与软更新策略，并完善日志与可视化管线。通过更稳定的超参与数据流，显著提升训练收敛性与可观测性，适配更长上下文与更严格的合规约束。修复“仅用两字符匹配”的局限，改为遍历长度集合  $U$ 。

- 
- 在不改变总体 SAC 框架的前提下，本版将“前缀/后缀命中”的固定两字符限制推广为“遍历 `data/word_length_sets.json` 提供的长度集合  $U$ ”的可变长度匹配；以“最长可用命中”为停表准则，统一 `raw_action` 与 `bigram` 的拓扑规则。
  - 该修复显著提升中文词法对齐与奖励密度，降低字符级训练中的信用分配难度，并改善 OOV（词汇库外）/ 长词边界的泛化。

关键词：中文知识蒸馏；可变长度后缀；最长可用命中；SAC；字符级策略

- v1.0.0 局限：前缀与后缀命中默认使用“两字符”作为判定窗口，无法适配多字词（如“面红耳赤”“如今”“未来”等），易造成早停或错停。
- v2.0.0 改进：从 `data/word_length_sets.json` 读取并集长度集合  $U$ （如  $U = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13\}$ ），匹配时按降序遍历  $U \cap [1..len(\cdot)]$ ，以“最长可用命中”为停表准则；前缀与后缀逻辑保持一致。

## 2 形式化定义

- 词表并集（只读）：

$$\mathcal{C} = \text{Catalog} = \text{chinese\_name\_frequency\_word.json} \cup \text{chinese\_frequency\_word.json}.$$

- 允许长度集合：

$$U = \text{union.lengths} \subset \mathbb{N}, \quad \text{来自 } \text{data/word_length_sets.json}.$$

- 目标字符： $\chi_t$ ；上一摘要预览： $\text{prev}_t$ （观测与渲染仅取末尾  $\leq 1$  字）。
- 源串：

$$\text{source}_t = \text{prev}_t \oplus \chi_t.$$

- 约定:

- $\text{tail}(s, L)$ : 串  $s$  的长度为  $L$  的后缀;  $\text{prefix}(s, L)$ : 长度为  $L$  的前缀;
- $\text{len}(s)$ : 串长;  $\text{is\_cjk}(s)$ : 是否全为 CJK 统一表意字符。

## 3 可变长度“前缀命中”（历史左扩）

目标是使源串满足“存在长度  $L$  的前缀命中”:

$$\exists L \in U \cap [1..|\text{source}_t|], \quad \text{prefix}(\text{source}_t, L) \in \mathcal{C}.$$

若不成立，则沿“历史字符对”向左扩展  $\text{prev}_t$ , 至多  $N$  步 ( $N = \text{character\_history\_extension\_limit} = 16$ )，每步重检上述条件，并采用“最长命中”原则。

```
function EXTEND_PREV_FOR_PREFIX_HIT(prev, chapter, history_pairs, N, U):
    source = prev + chapter
    if len(source) < 1: return prev
    step = 0
    while (not ANY_PREFIX_HIT(source, U)) and step < N and history_pairs.has_left(step):
        pair = history_pairs.left(step) # head+tail, tail 与 prev 对齐
        if prev and not prev.startswith(pair.tail):
            step += 1; continue
        prev = pair.head + prev
        source = prev + chapter
        step += 1
    return prev

function ANY_PREFIX_HIT(source, U):
    for L in sort_desc(U n {1..len(source)}):
        seg = source[:L]
        if is_cjk(seg) and seg ∈ C: return True
    return False
```

注: v1.0.0 的“两字符前缀”可视为固定  $U = \{2\}$  的特例；本版改为一般  $U$ 。

## 4 可变长度“后缀命中”（raw\_action 与 bigram）

### 4.1 raw\_action 的后缀拓扑

以策略首字  $c$  为起点，逐个追加未来真值字符形成  $q$ ；遇“后缀命中”即停，优先最长  $L$ :

$$\exists L \in U \cap [1..|q|], \quad \text{tail}(q, L) \in \mathcal{C}.$$

```

function EXTEND_RAW_ACTION_SUFFIX(c, future_chars, U):
    q = dedup_head_repeat(c) # 处理首字重复, 如“辑辑...”→“辑...”
    for ch in future_chars:
        q += ch
        for L in sort_desc(U ∩ {1..len(q)}):
            seg = tail(q, L)
            if is_cjk(seg) and seg ∈ C:
                return q, seg, annotate(seg) # 命中即停 (最长优先)
    # 未命中: 回退到最长连续 CJK 尾段, 仅用于日志可读性
    return q, longest_cjk_tail(q, max(U)), annotate_optional()

```

## 4.2 bigram 的前向拓扑

令  $s = \chi_t \oplus q$ , 在  $U$  上做“后缀命中”, 命中即停; 用于 `bigram` 奖励与注记:

```

function FORWARD_EXTEND_BIGRAM(chapter, q, future_chars, U):
    s = chapter + q
    best2 = tail(s, min(2, len(s))) # 仅用于未命中回退的显示
    for ch in future_chars:
        s += ch
        for L in sort_desc(U ∩ {1..len(s)}):
            seg = tail(s, L)
            if is_cjk(seg) and in_catalog(seg):
                return seg, s
    return best2, s

```

bigram 注记以“后缀命中”为核心, 如 `data/chinese_frequency_word.json\#236703`。

## 5 奖励构成 (与 v1.0.0 兼容)

质量/词法/洁净度分量保持不变:

$$\mathcal{N}_\gamma(x) = 1 - (1 - x)^\gamma, \quad S_t = Q_t + L_t - P_t.$$

字符模式加成保留, 但“命中”由  $U$  上的后缀决定:

$$\delta_t = \begin{cases} 1.0, & \exists L \in U, \text{tail}(s, L) \in C, \\ 0.5, & \text{tail}(s, 1) = \text{target\_char}, \\ 0, & \text{otherwise.} \end{cases}$$

$$B_t^{\text{char}} = B_t + 0.5 \chi_t^{\text{soft}} + \delta_t, \quad \Delta_t^{\text{char}} = \Delta_t + 0.25 \chi_t^{\text{soft}}.$$

# 6 复杂度与实现考量

- 命中判定：每次  $O(|U|)$ ，实际  $|U| \approx 10-12$ ，结合缓存可控。
- 历史左扩：最坏  $O(N)$ ， $N = \text{character\_history\_extension\_limit} = 16$ 。
- 相较 v1.0.0：开销由常数 2 升为  $|U|$ ，但命中率与对齐质量显著提升。

## 7 与 v1.0.0 差异清单

- 前缀命中：由“固定两字”→“ $U$  上最长前缀命中”。
- `raw_action` 后缀：由“固定两字”→“ $U$  上最长后缀命中（命中即停）”。
- `bigram` 后缀：同 `raw_action`，复用  $U$ 。
- 日志注记：显示命中词及来源（如 `data/chinese_frequency_word.json\#236703`），可附命中长度  $L$ 。
- 失败回退：保留“last-2 展示”仅用于可读性，不再作为命中判据。

## 8 兼容性与落地

- 配置项不变：`character_history_extension_limit` 仍为 16；仅匹配逻辑改为使用  $U$ 。
- 数据依赖：`data/word_length_sets.json` 必须存在且包含 `union.lengths`；缺失时建议回退  $U = \{2..8\}$  并记录告警。
- 知识蒸馏： $U$  来自真实语料词长分布，将“词级先验”蒸馏到字符级策略，提升样本效率与稳定性。

## 9 评测建议

- 消融：固定  $U = \{2\}$  vs.  $U = \text{union.lengths}$ ；比较收敛速度、最终 reward、词法命中率。
- 命中统计：平均命中长度、命中率（前缀/后缀分别统计）。
- 泛化：跨域/OOV（词汇库外）文本的表现差异，长词边界识别能力。

附录：记号

- $\mathcal{C}$ : 词表并集； $U$ : 长度集合； $\chi_t$ : 目标字符； $\text{prev}_t$ : 摘要预览； $\text{source}_t$ : 源串； $q$ : `raw_action` 串； $s$ : `bigram` 串；
- $\text{tail}(s, L)$ : 后缀； $\text{prefix}(s, L)$ : 前缀； $N$ : 历史左扩上限； $\text{is\_cjk}(\cdot)$ : CJK 判定。

---

### 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。