December 23, 2025

Design choices

- **Self-certainty per semantic unit (fixed).** Use KL-from-uniform token-wise certainty and average within the unit (as above). Equivalent CE form (differs by a constant):

$$c^i_{t,m} \equiv -\frac{1}{|\mathcal{V}|}\sum_{j\in\mathcal{V}} \log p(j \mid \theta, \theta^i_{t-1}, y^i_{t,1:m-1}) \quad (+\text{const}).$$

- **(Optional) TransformSC (stability knob).** Choose one:
  - *Centering:* $\widetilde{\mathrm{SC}}^i_t \leftarrow \mathrm{SC}^i_t - \frac{1}{N_{\text{alive}}}\sum_{r=1}^{N_{\text{alive}}} \mathrm{SC}^r_t$.
  - *Clipping:* $\widetilde{\mathrm{SC}}^i_t \leftarrow \mathrm{clip}(\mathrm{SC}^i_t, [a, b])$.
- Increasing annealing $f_t(t) = \beta_t$ (choose one).
  - (A) *Linear ramp:* $\beta_t = \min(1,\ t/T_{\text{anneal}})$.
  - (B) *Power ramp:* $\beta_t = \min(1,\ (t/T_{\text{anneal}})^\gamma),\ \ \gamma \geq 1$.
  - (C) *Saturating ramp:* $\beta_t = 1 - \exp(-\kappa t)$.
  - (D) *Strong default (ESS-targeted ramp):* choose the smallest $\beta_t \geq \beta_{t-1}$ such that $\mathrm{ESS}_{\text{win}}(\beta_t) \geq \rho\, N_{\text{alive}}$ (solve by bisection), where $\mathrm{ESS}_{\text{win}}(\beta)$ is computed by re-evaluating $w^i_{t'}(\beta) = \exp(\beta\, \widetilde{\mathrm{SC}}^i_{t'})$ for the rows in the current window.
- **Sliding window k_t (constant or adaptive; you asked both).**
  - *Constant:* $k_t \equiv k$.
  - *Adaptive by time:* $k_t = \min(k_{\max},\, t)$.
  - *Adaptive by N:* $k_t = \min\big(k_{\max},\, \lceil c \log N_{\text{alive}} \rceil,\, t\big)$.
- **(Your #3) Resample condition (choose one; ESS is the principled default).**
  - (Primary) Resample if $\mathrm{ESS}_{\text{win}} < \tau\, N_{\text{alive}}$ (e.g. $\tau = 1/3$).

Algorithm 1 Persistent Sequential Monte Carlo with Self-Certainty (Sentence/Semantic-Unit Time)

Data:
Number of particles N.
Question prompt θ.
Annealing function $f_t(\cdot)$.
LLM $\pi(\cdot)$.
Self-certainty scorer $\mathrm{SC}(\cdot)$.
Sentence list $lst[\ ]$.
Sliding-window length rule k_t (constant k or adaptive).
Score matrix W of shape $k_t \times N$ storing windowed, scaled weights.

Result: N completed samples (sentences/trajectories) in $lst[\ ]$.

```
t ← 1, N_alive ← N, W ← [ ]                                  ▷ W stores rows w̃_{t'}^{1:N} for recent t'
for i = 1 to N_alive do
    θ_1^i ∼ π(θ)                  ▷ Generate 1 semantic unit conditioned on prompt; cache token probs for SC
end for
while N_alive > 0 do
    β_t ← f_t(t)                                              ▷ Increasing score scale / annealing
    for i = 1 to N_alive do
        Let the t-th semantic unit in θ_t^i have tokens y^i_{t,1:n_t^i}.
        Compute average self-certainty for this semantic unit:
        SC_t^i ← −(1/(n_t^i |V|)) Σ_{m=1}^{n_t^i} Σ_{j∈V} log( |V| · p(j | θ, θ_t^i, y^i_{t,≤m−1}) )
        (Optional) S̃C_t^i ← TransformSC({SC_t^r}_{r=1}^{N_alive}, SC_t^i)     ▷ e.g., centering/clipping
        Define positive "likelihood proxy" L_t^i ← exp(S̃C_t^i).
        Scaled window-weight (two equivalent parameterizations):
            (A) w̃_t^i ← (L_t^i)^{β_t} = exp(β_t S̃C_t^i)                       ▷ recommended
            (B) w̃_t^i ← β_t · L_t^i                                           ▷ simpler, but less stable
    end for
    k_t ← ChooseWindow(k, t, N_alive)                                         ▷ constant or adaptive
    W ← W.append(w̃_t^{1:N_alive}); if |W| > k_t, then W ← W.removeOldest().
    Z_norm ← Σ_{all entries w∈W} w;   W_norm ← W/Z_norm.
    ESS_win ← ( Σ_{all entries u∈W_norm} u^2 )^{−1}.
    if ResampleCondition(ESS_win, N_alive, k_t) then
        {θ_t^i}_{i=1}^{N_alive} ∼ resample( {{θ_{t'}^i}_{i=1}^{N_alive}}_{t'=t−k_t+1}^{t}, W_norm )   ▷ sample N_alive trajectories from the window-pool
        (Optional) W ← [ ]                           ▷ reset window after resample (recommended option)
    end if
    for i = 1 to N_alive do
        θ_{t+1}^i ∼ π(θ_t^i)        ▷ Generate next semantic unit conditioned on current trajectory; cache token probs
    end for
    for i = 1 to N_alive do
        if token_end ∈ θ_{t+1}^i then
            lst.append(θ_{t+1}^i)
            remove particle i from alive set; N_alive ← N_alive − 1
        end if
    end for
    t ← t + 1
end while
```

Key formulas:

$$\mathrm{SC}_t^i \leftarrow -\frac{1}{n_t^i\,|\mathcal{V}|}\sum_{m=1}^{n_t^i}\sum_{j\in\mathcal{V}}\log\Big(|\mathcal{V}|\cdot p(j \mid \theta,\, \theta_t^i,\, y^i_{t,\le m-1})\Big)$$

$$\widetilde{\mathrm{SC}}_t^i \leftarrow \textsc{TransformSC}(\{\mathrm{SC}_t^r\}_{r=1}^{N_{\text{alive}}}, \mathrm{SC}_t^i), \quad \mathcal{L}_t^i \leftarrow \exp(\widetilde{\mathrm{SC}}_t^i)$$

$$\text{(A) } \tilde{w}_t^i \leftarrow (\mathcal{L}_t^i)^{\beta_t} = \exp(\beta_t \widetilde{\mathrm{SC}}_t^i), \qquad \text{(B) } \tilde{w}_t^i \leftarrow \beta_t \cdot \mathcal{L}_t^i$$

$$Z_{\text{norm}} \leftarrow \sum_{\text{all entries } w\in W} w; \quad W_{\text{norm}} \leftarrow W/Z_{\text{norm}}, \qquad \mathrm{ESS}_{\text{win}} \leftarrow \Big(\sum_{\text{all entries } u\in W_{\text{norm}}} u^2\Big)^{-1}$$

$$\{\theta_t^i\}_{i=1}^{N_{\text{alive}}} \sim \textsc{resample}\Big(\{\{\theta_{t'}^i\}_{i=1}^{N_{\text{alive}}}\}_{t'=t-k_t+1}^{t},\ W_{\text{norm}}\Big)$$