Stage-11 Text ARC — Steps 1–10: Findings & Final-Bench Readiness

Owner: You
Period: Stage-1 → Stage-11 (Steps 1–10)
Models: toy → GPT-2 (dev harness)
Runner: `text_arc_unified.py` (one script, profiles supported)

---

0) Executive Summary

- We integrated **Steps 1–6** into a single decoding path ("Geo") and validated it against Stock.
- We added **Steps 7–10** (metrics export, perf profiles, dtype/compile support, single-runner consolidation).
- **Quality:** Geo consistently reduces repetition/loopiness vs Stock, with similar output lengths. Post-tweak (**v4b tap-9**) we observe healthier 2–3-token bursts and fewer micro-loops.
- **Ablation:** Turning the **soft denoiser off** increases duplication/loopiness without improving content.
- **Perf:** Swapping to **fp16/bf16** preserves text quality; tokens/sec improves depending on GPU profile.
- **Readiness:** We are ready to run final benchmarks. Optional refinements (true trend @ tap; tap-true PCA) can become **v4c** later.

---

1) What changed by step

Steps 1–4

Goal: Always-on geometric warp with a soft trend gate; unify in one script. - **Warp (α_min > 0):** Always applies a small inward step toward a geometric center (EMA + optional PCA-2 plane). - **Soft trend gate (`g_tr`):** Smoothly scales the step; latch/linger avoids chattery on/off. - **Single runner:** Stock vs Geo, same prompts/IO (JSONL). Telemetry optional.

Steps 5–6

Goal: Add **Detect (gain-only)** + **Soft Denoiser**. - **Detect (gain-only):** Matched filter + null calibration → a *gain* factor (`g_det`) that **never disables** warp; it only amplifies when evidence is real. - **Soft Denoiser:** EMA + short median buffer + jitter blend + phantom-guard; *sign-safe* so it can't reverse inward direction. - **TweakA** (param patch): Longer, smoother bursts (≈2–3 tokens), lower duplication, same lengths.

Steps 7–10

Goal: QA & operations. - **7 – Denoiser ablation:** Toggle `--use_denoise 0` to verify improvement with denoiser ON. - **8 – Metrics export:** `--metrics_json / --metrics_csv` produce run-level stats (prompts, new tokens, mean `s` / `g_det` / `alpha`, burst stats; plus `elapsed_sec`, `tokens_per_sec`). -

9 – Perf profiles: `--perf_profile`, `--dtype {auto,fp16,bf16}`, `--compile 1` to sweep throughput. - **10 – Consolidation:** Keep everything in **one script**; configs via `--config` and frozen via `--save_config`.

---

2) Data & artifacts reviewed

Runs (all JSONL, same prompt family): - **Geo v4b tap-9**: `/mnt/data/generations_geo_steps.v4b.tap9.jsonl` - **Geo, no-denoise (ablation)**: `/mnt/data/generations_geo_steps.v4b.tap9.no_denoise.jsonl` - **Geo, fp16 sweep (T4)**: `/mnt/data/generations_geo_steps.v4b.tap9.t4_fp16.jsonl` - **Stock baseline**: `/mnt/data/generations_stock.v4b.tap9.jsonl`

Telemetry (optional): `/mnt/data/geo_steps1_6.v4b.tap9.telemetry.jsonl`

One-file summary: - CSV: `/mnt/data/steps7_10_summary.csv` - JSON: `/mnt/data/steps7_10_summary.json`

Runner & profiles: - Runner: `/mnt/data/text_arc_unified.py` - Profile (Geo v4b tap-9): `/mnt/data/calib/profile_v4b_tap9_text.json`

---

3) Measurement approach

Text quality proxies (no truths required): - **Lengths:** avg/median words per completion (guard against truncation). - **Duplication:** adjacent duplicate fraction; repeated 3-grams. - **Loopiness:** token runs ≥ 6 or repeated 4-grams (naïve loop heuristic). - **Diversity:** unique-word ratio.

Telemetry (Geo mode): per token `alpha`, `s`, `g_tr`, `g_det`, `radius`, `step_norm`. - **Health signatures:** - **Bursts**: `s` sustained for ~2–3 tokens (not 1-token blips). - **Inward progress**: `radius` shrinks on average. - **Denoiser effect**: `step_norm` spikes are tamed during wobble.

Perf: `elapsed_sec`, `tokens_per_sec` from the runner metrics; dtype/compile/profiles toggled.

---

4) Findings

4.1 Geo vs Stock (quality)

- **Lower repetition/loopiness** with **similar lengths** in Geo → cleaner completions without harming coverage.
- Telemetry shows **healthy 2–3-token bursts** and positive average **radius shrink**, consistent with the intended geometry.
- Expected scoreboard outcome once truths are applied: a **small but real uplift** on accuracy/F1, especially on wobble-prone items.

4.2 Denoiser ablation (Step-7)

- With **denoiser OFF**, we see **higher loopish/dup rates** at similar lengths.
- With **denoiser ON**, spikes in `step_norm` are reduced; outputs are steadier; no evidence of over-smoothing.

4.3 Detect gain behavior

- `g_det` shows **peaks aligned to progress**, not sticky; combined with latch/linger this avoids one-token "blips."
- Null calibration stabilizes the baseline; micro-peaks are tamed in the **TweakA** profile.

4.4 Perf sweep (Step-9)

- **dtype changes (fp16/bf16)** preserve text quality in our checks; use GPU profiles for throughput gains.
- Use `tokens_per_sec` (metrics JSON) to confirm speed-up on your hardware.

4.5 Reproducibility & ops

- **Profiles** (`--config`) and **frozen configs** (`--save_config`) provide exact reruns.
- Telemetry + metrics give **explainability** (why a run won/lost) without over-engineering the pipeline.

---

5) Recommended profile for final bench

- **Geo: v4b tap-9** (TweakA) — `/mnt/data/calib/profile_v4b_tap9_text.json` (Detect gain-only; Soft Denoiser on; `linger↑`, `trend_tau↓`, `detect_width↓`, `null_K↑`, `k_det↓`, `denoise_k↓`, `denoise_tau↑`, `denoise_window↑`.)
- **Stock:** no warp (baseline) — run from the same runner with `--gen_mode stock`.

Commands

```
# GEO (metrics + telemetry optional)
python3 text_arc_unified.py
  --config calib/profile_v4b_tap9_text.json
  --prompts calib/ngf_eval_prompts_60.txt
  --metrics_json metrics_geo.v4b.tap9.json
  --out generations_geo_steps.v4b.tap9.jsonl

# STOCK baseline
python3 text_arc_unified.py
  --gen_mode stock
  --prompts calib/ngf_eval_prompts_60.txt
  --metrics_json metrics_stock.v4b.tap9.json
  --out generations_stock.v4b.tap9.jsonl
```

6) What we deliberately did not add (yet)

To avoid over-engineering right now: - **True trend @ tap** (radius-decay) — would make `g_tr` more physically grounded; safe as a later **v4c**. - **Tap-true PCA calibration** — higher fidelity to the chosen layer; also a clean v4c addition.

These are toggles we can add post-bench if the A/B suggests non-trivial gains (>~1–2 pts or improved stability on long prompts).

---

7) Risks & mitigations

- **Detect too peaky** on some prompt families → mitigate with `null_q↑` (e.g., +0.01) or `detect_width↓` (e.g., 24→20).
- **Chattery bursts** (mean burst <2) → `linger↑` by +1 or `trend_tau↓` by 0.02–0.03.
- **Muted outputs** (over-cautious) → `alpha0↑` slightly (e.g., +0.01) or `null_q↓` by 0.01.
- **Throughput variance** → use `--perf_profile` + dtype/compile flags and record `tokens_per_sec` in metrics.

---

8) Decision & next steps

- **Decision:** Proceed to final benchmarking with **Geo v4b tap-9** vs **Stock** on the full prompt set.
- **Next:** 1) Fill `truths.csv` (adjudication sheet available) and run the A/B scorer. 2) Lock **v4b tap-9** as submitted; preserve `--save_config` output. 3) Optional: run a small A/B with **v4c** (true trend + tap-PCA) and promote only if ≥ +1–2 pt.

---

9) Appendix — handy artifacts

- Runner: `/mnt/data/text_arc_unified.py`
- Profile (Geo v4b tap-9): `/mnt/data/calib/profile_v4b_tap9_text.json`
- Geo (v4b tap-9): `/mnt/data/generations_geo_steps.v4b.tap9.jsonl`
- Geo (no-denoise): `/mnt/data/generations_geo_steps.v4b.tap9.no_denoise.jsonl`
- Geo (fp16): `/mnt/data/generations_geo_steps.v4b.tap9.t4_fp16.jsonl`
- Stock: `/mnt/data/generations_stock.v4b.tap9.jsonl`
- Telemetry: `/mnt/data/geo_steps1_6.v4b.tap9.telemetry.jsonl`
- Summary: `/mnt/data/steps7_10_summary.csv`, `/mnt/data/steps7_10_summary.json`