🎨Gemini Prompt Optimization Guide

Character Pose Generation & Appearance Consistency

> Working notes for Little Hero Books – improving 12-pose consistency and locking character appearance.

⛳Principles

1) **Image > Text.** Provide visual control sources in a strict order. 2) **Determinism where possible.** Same parts, same order, same settings. 3) **One change per test.** Measure, keep wins, revert losses. 4) **Short, unambiguous language.** Avoid poetic phrasing. 5) **Audit everything.** Hashes for inputs/outputs; record config.

Standard Parts Order (every call)

1. **STYLE_BOARD** — a style sheet image that defines line weight, palette, textures, and rendering style.
2. **BASE_CHARACTER** — the child's locked appearance reference (frontal or neutral angle), generated earlier for this characterHash.
3. **HAIR_CHIP** — a cropped patch with the exact hairstyle silhouette in opaque mass (no flyaway gaps); use same hair chip across all poses.
4. **POSE_REF** — the pose-only reference (no clothing or identity cues); can be line art or silhouette.
5. **TEXT** — minimal instruction to combine A (appearance) with B (pose).

> **Always keep this order** to help the model resolve conflicts consistently.

System Prompt (fixed across all 12 poses)

Role: You are a rendering assistant. Combine APPEARANCE (A) with POSE (B) without changing A.

Rules: - A = appearance (face, hair, skin tone, clothing colors & logos, proportions, style) from STYLE_BOARD + BASE_CHARACTER + HAIR_CHIP. - B = body posture and limb orientation from POSE_REF only. - If A and B conflict, **follow A** and reinterpret B's limb orientation to match A's anatomy. - **Do not change** hair silhouette, facial features, clothing logos or colors. - Output: a single, front-lit illustration in the STYLE_BOARD style. Clean alpha, **no halo/pinholes** in hair; opaque hair mass.

Trait Manifest (embed in User text)

Provide a compact, machine-readable block to “lock” traits. Include a checksum for BASE_CHARACTER so the model treats A as authoritative.

```
TRAIT_MANIFEST = {
  "character_hash": "{characterHash}",
  "base_image_sha256": "{baseSha256}",
  "hair": {"style": "{hairStyle}", "color": "{hairColor}", "silhouette":
"opaque_mass"},
  "skin_tone": "{skinTone}",
  "eyes": {"color": "{eyeColor}", "shape": "{eyeShape}"},
  "clothes": {"top": "{topDesc}", "bottom": "{bottomDesc}", "palette":
"{palette}"},
  "style": {"line": "{lineWeight}", "texture": "paper_grain", "palette":
"warm_muted"}
}
```

Keep the object small and consistent. The goal is signaling, not verbosity.

User Prompt Template (per pose)

Context images in order: STYLE_BOARD, BASE_CHARACTER, HAIR_CHIP, POSE_REF

Text:

```
Use A (appearance) from STYLE_BOARD + BASE_CHARACTER + HAIR_CHIP.
Use B (pose) from POSE_REF.

Follow TRAIT_MANIFEST exactly. Do not add or remove accessories. Do not change
hair silhouette or clothing palette. If B conflicts with A, keep A and
reinterpret B.

Render a single child in the STYLE_BOARD style, clean alpha, solid hair
silhouette (no gaps), no background elements, no text.

[Insert TRAIT_MANIFEST JSON here]
```

Final reinforcement (last lines): - “Do not modify hair silhouette.” - “Do not modify facial features.” - “Do not modify clothing colors or logos.”

Generation Config Recommendations

- **Temperature:** 0.15 (lower if residual drift persists; try 0.1).
- **Top-P/Top-K:** keep defaults unless drift persists, then lower Top-P modestly.
- **Size:** 1024×1024 or your production size; keep constant.
- **Safety filters:** standard; avoid aggressive auto-cropping.
- **Retries:** if pose ignored, retry once with an overlaid minimal "pose skeleton" reference.
- **Serialization:** run poses sequentially to avoid cross-run variability.

Hair Silhouette Strategy

- Create a **HAIR_CHIP** PNG with clean, closed silhouette (no wisps).
- Enforce opaque mass in System + User text.
- Prefer frontal/¾ angle for BASE_CHARACTER so silhouette maps well across poses.
- Export with transparency and test Bria background removal; iterate until "no pinholes."

Pose Consistency Strategy

- Pose refs are **style-aligned**: simplified anatomy, same proportions as BASE_CHARACTER.
- If certain poses drift, add a light pose skeleton overlay.
- Teach the model "pose grammar": head tilt, shoulders, hips, weight distribution.
- Keep **A-B-A sandwich** test variant (BASE → POSE → BASE) only if it shows measurable gain.

🎖️ Test Plan (one variable at a time)

1. Add HAIR_CHIP + System rules → measure hair alpha cleanliness (pinholes, halos).
2. Add explicit negatives ("do not borrow style from B") → measure appearance drift.
3. Add Trait Manifest with checksum → measure face/clothes stability.
4. Lower temperature to 0.10 if needed → measure overall variance.
5. Optional: A-B-A sandwich or pose skeleton overlays → keep only if net-positive.

Metrics: per-pose pass/fail on (a) hair alpha, (b) pose match, (c) face/clothes lock. Track with your existing auditing (hashes + config).

🧩 n8n Wiring Hints

- Keep a **fixed parts array** for every Gemini call; never vary order.
- Serialize poses (SplitInBatches = 1) + small Wait between calls.
- Compute `baseSha256` of BASE_CHARACTER and inject into TRAIT_MANIFEST.
- Log generationConfig alongside A/B hashes per pose.
- On retry, reduce temperature by 0.05 and add pose skeleton overlay.

Quick Checklist

- [ ] Style Board present first
- [ ] Base Character fixed for characterHash
- [ ] Hair Chip included (opaque silhouette)
- [ ] Pose Ref clean (no identity hints)
- [ ] System = A over B on conflict
- [ ] User = trait manifest + final 3 rules
- [ ] Temp $\leq$ 0.15, serialized calls
- [ ] Audit: save hashes + config per pose

Use this guide as a living checklist while we iterate on the 12-pose pipeline.