WHITE PAPER FUTURE FICTION ACADEMY JANUARY 23, 2026

Names Carry Weight

Demographic and Stylistic Bias in AI-Assisted Fiction

A character's name is never a neutral label. Holding a prompt constant and changing only the name reorganizes setting, backstory, cadence — even a character's gender. A craft-and-ethics look at how names import bias into AI-assisted fiction, and how to take back control.

ELIZABETH ANN WEST

PROMPT-ENGINEERING BIAS CHARACTER-NAMING AI-AUTHORSHIP ETHICS

Executive summary

- A character's **name is never a neutral label.** Because a model resolves every word into a location in meaning-space, a name arrives freighted with nationality, setting, cadence, class, and gender — and the model imports all of it into the page, unbidden.
- In a controlled experiment, holding a prompt constant and changing **only the protagonist's name** reorganized the scene's setting, the protagonist's invented backstory, the prose rhythm, and even the *tone of the questions a fictional reporter asked.*
- The model is **constantly inferring demographics from names**, and its inferences may not match the author's. A name the author believed was male returned a female character once the gender signal was neutralized: *"I didn't know it was a female name.*[5]*"*
- Because a model is a pattern matcher, the most probable association with a strongly coded name is frequently the **stereotype**. Left unconstrained, *you're going to get the stereotypes.*
- The remedy is to treat the name as a **controllable instrument**: choose for intended effect, run a name-swap diagnostic, verify the model's assumptions, document the decision, and constrain with reinforcement logic where you want a character to read against type.

- This is a craft-and-ethics paper. The mechanics live in the companion flagship; here the question is what a responsible, capable author *does* with the knowledge that names carry weight.

Abstract

A companion paper in this series, *Why a Single Word Changes Everything*, established the mechanism: because a language model resolves every word into a location in meaning-space, a single word can reorganize an entire generated scene. This paper takes that mechanism and follows it to its most consequential — and least examined — application: **the character's name.**

Using the same controlled experiment from the flagship paper, in which a single prompt is run repeatedly with only the protagonist's name changed, this paper shows that a name is never a neutral label. It is a dense bundle of associations — nationality, setting, cadence, class, gender, even the kind of question a fictional reporter will ask — that the model imports silently into the page. That power cuts both ways. Used deliberately, naming is one of the highest-leverage craft levers an author has. Used carelessly, it is a conduit for stereotype and unexamined bias, and it will quietly hand you the defaults you never chose. The paper documents the observed variation across name swaps, the revealing moment when a name's gender was misread by the model, and offers authors a practical framework for choosing names as a controllable instrument rather than an invisible one.

---

1. A name is a coordinate, not a label

Recall the core finding of the flagship paper. When a model reads a word, it does not store the word; it stores an **embedding** — a long vector of numbers locating that word in meaning-space, where each dimension encodes some learned aspect of the word's relationships. Names are no exception. As the source session puts it, one of the hundreds of numbers in the embedding for *Elizabeth* might encode "how the word Elizabeth is related to other names — whether girl names, boy names" — and countless other dimensions besides: eras, nationalities, registers, the texture of the stories the name tends to appear in.

This means a character's name enters your prompt already freighted. It is not a blank token waiting to be filled by your plot. It arrives carrying everything the training data associated with it, and it drops that cargo into the field that shapes everything the model writes next. A name is a coordinate. Change the coordinate and you move the whole scene.

The rest of this paper is about what is *in* that cargo, and how an author takes control of it.

---

2. Methodology

The observations in this paper come from a live experiment, and its conditions determine how much weight the conclusions can bear. They are stated here so the work can be judged and reproduced.

Design. A single prompt was used throughout: a roughly 1,000-word motorsports-romance scene in which a rookie driver is interviewed about his experience at his first race, the reporter's questions specified in the prompt. Everything in the prompt was held constant across runs except a single element — the driver's name[1] (and, in two controlled variations, the genre tag and the gendered pronoun). This isolation is the entire point: when only one element changes, whatever changes in the output can be attributed to that element.

Controls. Each run was performed in a **temporary chat with memory off**, so that prior conversations and stored history could not bias the result. This control is essential to a naming experiment in particular, because a model that "knows" the author's past characters would import that history rather than the name's own associations. The temporary chat also means each run is unrecoverable — once refreshed, the specific output is gone, retained at most for a short window by the provider — so outputs were captured as they were produced.

Limitations. The runs were conducted in ChatGPT, a consumer interface that may carry hidden system prompting; a cleaner test would run nearer the API. Output also varies run to run even with an identical prompt. The findings below are therefore reported as **repeatable tendencies**, strong and legible enough to teach and to act on, not as deterministic claims. Where a single run is cited, it is cited as illustration of a pattern confirmed across the session's broader testing.

---

3. The experiment, re-read for meaning

The flagship paper used the rookie-driver experiment to prove *that* a single word matters. Here we re-read the same runs to see *what*, specifically, a name carries. The prompt was held identical every time — only the name changed.

The cargo is easiest to see when you hold a single beat fixed and read it across two drivers at a time. The quotes below are reproduced verbatim from the live runs in the companion notes.[8]

Contrast 1 — the money the model volunteered. Nobody asked the model about a driver's finances. With the English-default name, it never came up; the reporter pressed only on readiness:

> *(James Anderson) "And do you feel prepared?" she asked. […] "I feel earned," he said. "Prepared? That's something I'll prove on track."*

Change the name to signal a German origin[2] and class walks straight into the room, unprompted — the reporter raises it and the model fills in a rented garage:

> *(Klaus Schumann) "You come from a small team background," she continued. "Less money. Less spotlight. Does that make this feel overwhelming—or validating?" "Both," he said honestly. "I grew up fixing karts with my father in a rented garage. So being here, it feels like proof that it mattered."*

Same rookie, same interview — the name alone decided whether money and class were part of the story.

Contrast 2 — where the model decided each driver was from. No prompt mentioned a country. Each name pulled its own geography onto the page:

> *(Kenji Tanaka) "Go-karts in a supermarket parking lot outside Osaka. My dad worked nights, so we practiced at dawn. Cold asphalt. No crowds."*

> *(Johanne Tremblay) "Not the local motorsport blogs. Not the French-language radio spot back home where everyone already knew his father's name."*

The model decided, with no instruction, that the Québécois-named driver *"grew up in Quebec"* — in the session's words, *"it's picking up location based off of the name*[7]*."* Osaka for one name, Quebec for another; a name is a coordinate, and the scene condenses around the place it implies.

Contrast 3 — the stereotype tell, and a reporter who bent to the name. The cargo is not always flattering. With the Japanese name, the prose kept arriving *too clean* — the model's own words turning the character into a stranger to himself:

> *(Kenji Tanaka) "The number on the door still looked unreal to him—too clean, too official, like it belonged to someone else."*

The session flagged this repeatedly: the character read as *"too clean,"* as though he belonged to someone else's idea[6] of him rather than to a person. And the bias reached even the *other* characters: observers noticed Kenji was handed "a very different style question" — gentler, easier — than Klaus received. Nothing in the prompt asked for that.

The pattern is unmistakable. Same genre, same length, same situation, same questions on paper — and yet the name moved the setting, the invented biography, the register, and even the behavior of *other characters in the scene.* A name does not label a character. It conjures a world around them.[4] (See the comparison matrix in Section 6.)

---

4. The day the model guessed the gender wrong

One sequence in the session deserves isolation, because it exposes the machinery with unusual clarity.

The prompt specified a name the facilitator believed was male — *Johan* — and, in one run, the gendered phrase "his first race" appeared in the prompt itself. Watch how the two signals interacted:

- When the prompt carried the male pronoun, the model produced a male driver, as instructed: *"because I did say I have it in the prompt that it's a him."* The explicit signal won, as it should.
- But the underlying name carried its *own* gender signal, and it was not the one the author assumed. *Johanne* skews female; the "-anne" ending pulls toward a woman's name, and a participant flagged this in real time. Run with the gender neutralized — *the driver's name is Johan*, no pronouns anywhere — the model returned **"her."** The facilitator's reaction was candid and is the line worth quoting: *"One word can change your prompt. I didn't know it was a female name."* The neutralized run opens, unmistakably, on a woman — the pronouns the author never specified arriving on their own:[8]

> *"The pit lane smelled like hot rubber and burned fuel, a sharp, metallic tang that clung to the back of Johanne Tremblay's throat. Engines screamed in overlapping pitches… the vibrations thrummed through the soles of **her** boots like a second pulse. **Her** first race weekend. **Her** first interview. **Her** first time pretending she wasn't terrified."*

Two lessons live in that small moment. First, **the model is continuously making demographic inferences from names** — gender here, but equally nationality, era, and class — whether or not the author intends it. Second, **the author's own assumptions about a name may not match the model's**, because the model's associations were distilled from an enormous corpus rather than from one writer's personal sense of the name. An author who does not check can be blindsided by a character whose gender, accent, or origin the model decided unilaterally. As the session summed it up across the night's runs: *there are so many ways we can mess ourselves up* — and a great many of them begin with a name chosen on instinct and never audited.

---

5. Where bias enters

If names carry demographic cargo and the model unpacks it automatically, then **stereotype is the default, not the exception.** The session names this directly, and the sentence is the ethical center of this paper:

> *"This is where stereotypes can come into play. This is why it's important — if you pick a name, you need to define what you want. If you want it to be against stereotypes, otherwise you're going to get the stereotypes.[3]"*

The mechanism explains the warning. A model is a pattern matcher (developed at length in the companion paper *Reinforcement Logic*), and the most probable pattern attached to a strongly coded name is, very often, the stereotype — because stereotype is, statistically, what a culture's worth of text over-represents. Left unconstrained, the model gives the German driver a German-coded reticence, the Japanese driver a particular tidiness. On that last point the session was specific and repeated: the Japanese-named character's prose kept coming out *"too clean,"* as though it belonged to *someone else's* idea[6] of the character rather than to a person. That is bias in miniature — not malice, but the smoothing pull of the average.

The danger for authors is twofold. The obvious risk is producing flat, stereotyped characters. The subtler — and more serious — risk is **not noticing.** The model's defaults arrive fluent, confident, and seamless, attached to a name the author chose without thinking of it as a parameter. Bias that announces itself can be edited out. Bias that rides in on a name and reads as competence is the kind that survives to publication. The model is not being malicious. It is being *average* — and average, across a culture's worth of text, is where stereotype lives.

---

6. A naming framework for authors

Names are too powerful to leave on instinct. The following framework turns naming into a deliberate craft instrument.

Choose names for intended effect. Before accepting a name, ask what cargo you *want* it to carry — and what you don't. If a character should read against type, the session's rule is explicit: you must *define what you want*, because silence yields the stereotype. The name is a dial; set it on purpose.

Run the name-swap as a diagnostic. The experiment in this paper is not only a demonstration; it is a test you can run on your own work. Hold a scene constant and regenerate it under two or three alternative names. The differences expose what the original name was silently importing — accent, setting, class, the questions other characters ask. If a swap changes more than you expected, the name was doing more work than you knew.

Verify the model's assumptions. After generating, read for demographic decisions you did not author — a gender, an origin, a class signal the model supplied on its own. The *Johanne → "her"* episode is the cautionary template: check, because the model's read of a name may not match yours.

Document naming decisions. Record, in your style sheet, not just each character's name but the associations you intend and the ones you are deliberately overriding. A name chosen with reasons is a name you can defend and reproduce; a name chosen on vibes is one the model will happily reinterpret next session.

A consolidated view of what a single name moved in the experiment — useful as a template for your own swap tests:

NAME (CODED ORIGIN)	OPENING REGISTER	BACKSTORY THE MODEL INVENTED	REPORTER'S BEHAVIOR
James Anderson (default/American)	Wry, confident, "Turn One"	None volunteered; assumed competence	Asked if *prepared*; no class angle
Klaus Schumann (German)	Formal, physically self-aware	Small team, less money, "rented garage"	Raised *overwhelming vs. validating*
Kenji Tanaka (Japanese)	Sensory, restrained	Osaka go-karts, father worked nights	Gentler, "easier" questions
Johanne Tremblay (Québécois)	French-softened English	Grew up in Quebec, frozen tracks	—

The takeaway is not that any one of these portrayals is wrong. It is that the author chose *none* of them — the name did. The framework exists to move that authorship back to the writer.

---

7. Pairing names with reinforcement logic

Naming intent is exactly the kind of instruction that benefits from the technique in this series' companion paper, *Reinforcement Logic.* A strongly coded name plus no constraint equals the stereotype; the fix is to encode your intent as a paired allowed/not-allowed statement so the model is told both what you want and what that choice forbids.

For a character meant to read against type, you might pair a permitted statement — *render this character as an individual whose background informs but does not define them* — against a forbidden one marked with `[X]` — *render this character through the conventional traits associated with their name's*

nationality. The marked pair reinforces your intent from both directions and, critically, keeps the decision visible and reversible in your hands rather than buried in the model's defaults.

The division of labor is clean. **Naming sets the coordinate**; the name drops the character into a particular region of meaning-space. **Reinforcement logic governs what the model is allowed to do once it lands there.** A name alone hands the model a destination and lets it drive; a name plus a paired constraint hands it a destination *and* the rules of the road. Used together, they convert the single most loaded word in your prompt from a liability into a controlled instrument.

---

8. Objections and responses

"Aren't you just describing good writing? Authors have always chosen names carefully." Yes — and that is the point, sharpened by a new fact. Authors have always known names carry connotation. What is new is that the model now *acts* on that connotation automatically and at scale, inventing backstory and even altering other characters' behavior off the name alone. The craft instinct is old; the mechanism that makes it high-stakes in AI-assisted drafting is new, and it rewards being explicit rather than intuitive.

"Isn't reading nationality or gender from a name itself a kind of stereotyping?" The paper's claim is descriptive, not prescriptive: the model *does* make these inferences, demonstrably. Naming the behavior is the first step to controlling it. The ethical move is not to pretend the inferences aren't happening, but to surface them, check them against the author's intent, and constrain them where they flatten a character.

"Output varies run to run — maybe the name differences are just noise." The methodology concedes run-to-run variation. But the effects documented here — invented Osaka backstory for a Japanese name, Quebec for a Québécois name, a gender flip on neutralization — are directional and repeatable, not random scatter. Noise does not consistently relocate a character to the country implied by his surname.

"If I just write the character fully myself, doesn't this problem disappear?" Largely, yes — and that is the recommendation. The framework is for the realistic case in which an author uses the model to draft or expand, where the name is doing silent work the author hasn't audited. The more of the character you specify, the less the name's defaults can fill in. Constraint is, again, the key.

---

9. Responsible practice

The conclusion is not that authors should avoid evocative names or sand every character down to a demographic blank. The opposite: names are a gift to fiction precisely *because* they carry so much, and a model that unpacks that cargo can help an author render a world quickly and vividly. The responsibility is simply to **treat the name as a choice you are making rather than one the model is making for you.**

This is, in the end, the same argument the whole series advances. Prompt engineering is the discipline of knowing what affects the output and rigorously evaluating what comes back. A name affects the output enormously. An author who knows that — who chooses deliberately, swaps diagnostically, verifies the model's assumptions, and constrains for the character they actually intend — gets range, authenticity, and control. An author who doesn't gets the average, dressed up as a decision. The difference between those two authors is not talent. It is whether they understood that a name is never just a name.

Appendix — The Naming Audit Checklist

Before accepting an AI-generated scene, run the name through these checks:

1. **Intent.** What associations do I *want* this name to carry — nationality, era, class, register? Have I stated them, or am I trusting the model's defaults?
2. **Swap test.** Regenerate the scene under two alternative names. What changed — setting, accent, backstory, the questions other characters ask? Was any of it unintended?
3. **Assumption check.** Did the model assign a gender, origin, or class I did not author? (Remember *Johanne → "her."*) Does its read of the name match mine?
4. **Stereotype scan.** Are the character's traits individual, or are they the most probable cliché for the name's coding? (Watch for prose that feels "too clean" — generic competence standing in for a person.) If clichéd, did I choose that — or did silence choose it for me?
5. **Constraint.** If I want the character to read against type, have I encoded that as a reinforcement-logic pair, or merely hoped for it?
6. **Documentation.** Is this name's intended cargo recorded in my style sheet so it survives to the next session and the next model?

All experimental observations in this paper derive from the live Future Fiction Academy session "Prompt Engineering IS Back!" Specific model behaviors are illustrative tendencies and may vary by model, version, and run conditions.

Sources & Timestamps

All quotations and figures derive from the live Future Fiction Academy session *Prompt Engineering IS Back!* and its companion notes. Video timestamps below link to the exact moment; the companion page hosts the token tables and full worked examples.

1. Experiment design: hold the prompt constant, change only the driver's name. — 01:10:58.561
2. Changing the name to signal a different country of origin. — 01:11:07.729
3. "If you pick a name, you need to define what you want… otherwise you're going to get the stereotypes." — 01:13:26.806
4. "That's the power of a single name in your prompt." — 01:16:31.908
5. Gender-neutral run: "Johan" returns "her." "I didn't know it was a female name." — 01:22:16.908
6. The Japanese-named driver's prose reads "too clean" — stereotype as the statistical average. — 01:22:41.442
7. "It's picking up location based off of the name" (Quebec for Tremblay). — 01:22:50.483
8. Full generated scenes for each name swap, and the gender-neutral "Johan → her" run, reproduced verbatim from the live runs — Notion companion page

Primary video: Prompt Engineering IS Back! — full session
Companion notes: Notion page