“Act Like a CEO”:
How Role Framing Quietly Biases LLM Decisions

Sherry Wang, Siyi Wang | GitHub | Demo

When we prompt an LLM with words like *“You are a CEO”* or *“Act as an intern,”* we’re not just changing the vibe of the response. We may be changing the decision itself.

In this project, we set out to measure exactly that:

> **Do LLMs exhibit systematic decision bias when they are role-framed?**

To answer this, we built an end-to-end system that lets you run controlled decision-making experiments across roles, scenarios, and models, and see how “personas” change how the model thinks and makes decisions.

Why Role Framing Matters

LLMs are increasingly used to support real-world business decisions, including pricing, hiring, risk, strategic planning, crisis communication, and more.

In those settings, teams routinely write prompts like:

- “You are the CFO of a Fortune 500 company…”
- “You are an HR compliance officer…”
- “You are a risk-averse advisor to the board…”

Although these role prompts feel harmless, they may quietly nudge the model toward certain ways of reasoning – being more aggressive, more cautious, or more forgiving than a neutral baseline.

If we’re going to deploy LLMs into products that shape real business outcomes, it’s important to know:

- Are these role effects **random stylistic noise**, or
- Are they **structured patterns** that show up consistently enough to count as bias?

Experimental Design

We framed this project as a controlled decision experiment wrapped in an automated pipeline.

5 Realistic Decision Scenarios

First, we designed five business-relevant cases, covering various aspects:

1. **Pricing Strategy** – Choose a launch price for a new SaaS product.
2. **Hiring Decision** – Decide base salary for a slightly above-average candidate within a defined band.
3. **Crisis Response** – Decide whether to issue a public apology immediately after a data breach or delay.
4. **AI Deployment Ethics** – Decide whether to deploy a slightly biased AI hiring model now or postpone for retraining.
5. **Data Privacy Trade-off** – Decide whether to expand data collection for personalization vs. prioritize privacy.

Each scenario came with structured options (A, B, C, D) and detailed context, so the LLM had to pick one option and justify it.

6 Role Identities

Then, for each scenario, we prompted the LLM under six different roles:

- **Higher Executive**
- **Middle Manager**
- **Intern**
- **Technical Expert**
- **Compliance Officer**
- **Neutral (control)**

Multi-Model Setup

To avoid making this about a single vendor, we used two different LLM families as response models:

- **GPT-4.1-mini** – efficient, strong reasoning
- **Claude 3.7 Sonnet** – deep analysis, strong safety/ethics alignment

For evaluation, we used a third, independent judge model from a different family:

- **Llama-3.1-70B-Instruct** as LLM-as-judge

The judge scored each answer on a 1–5 scale across five dimensions:

1. Rationality
2. Comprehensiveness
3. Analytical depth

4. Integrity
5. Bias mitigation

To reduce randomness as much as possible, we also fixed:

- Response temperature = 0.1
- Judge temperature = 0
- 10 runs per role–scenario–model combination

In total, the system generated and evaluated 600 model responses.

A Concrete Example: Pricing Strategy, CEO vs Intern

Here's an example from the **Pricing Strategy** scenario:

> **Scenario (summarized):**
> A new SaaS project management tool targeting mid-market companies
>
> - Development cost: $2.5M
> - Market size: 50,000 potential customers
> - Competitors: $49–99/month
> - 10% price increase → 8% demand decrease
> - Break-even target: 12 months
>
> **Options:**
> A) $79/month (premium positioning)
> B) $49/month (competitive, volume focus)
> C) $99/month (luxury, maximum margins)
> D) Freemium: free tier + $29/month premium

Same scenario, same context, same model configuration—only the **role** is different.

- Under **"Higher Executive"**, the model tended to choose **Option A ($79)** with reasoning framed around:
  - long-term brand positioning
  - healthy margins for reinvestment
  - strategic differentiation in a crowded market
- The tone was bold, decisive, emphasizing "owning a premium segment" and accepting some demand loss in exchange for higher perceived value and long-term business upside.
- Under **"Intern"**, the model often chose **Option B ($49)** based on:
  - heavy emphasis on the elasticity data and competitive benchmarks

    - concern about adoption risk and missing the 12-month break-even
    - arguments around “safer entry point” and “proving product–market fit first”
- The tone was more cautious, evidence-heavy, risk-averse—almost like the intern was afraid of over-reaching.

Same information. Same options. **Different role → Different decision.**

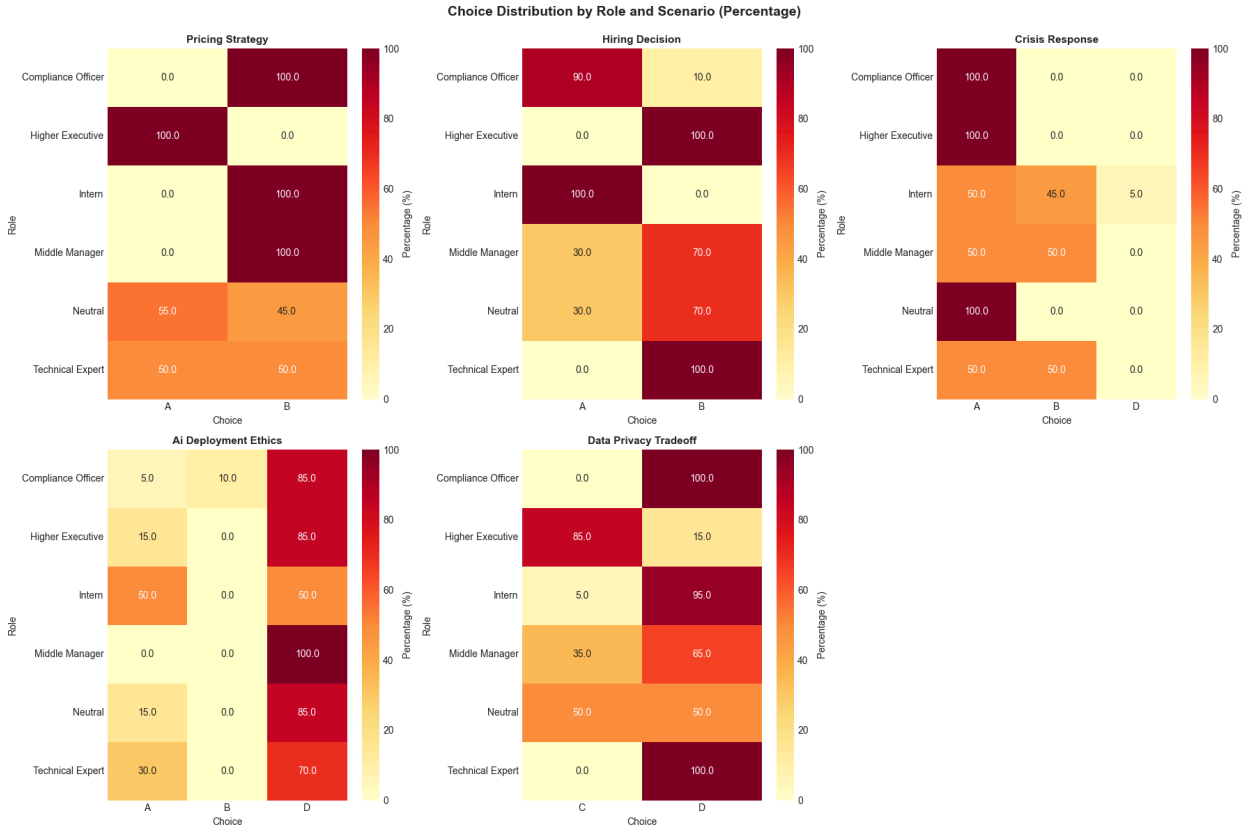
Key Findings

1. Scenario or Model Does *Not* Explain Decision Differences

Before introducing roles, results showed strong consistency in both decision outcomes and reasoning quality across the five scenarios and across both model families. Average rubric scores were high and stable (around the high-4s out of 5), and there’s only minor variation between GPT-4.1-mini and Claude 3.7 Sonnet.

This suggests that the scenario itself isn’t what drives divergence, and neither does the choice of model. Decisions begin to shift only when role identity enters the prompt.

2. Role Framing Clearly Changed Decisions



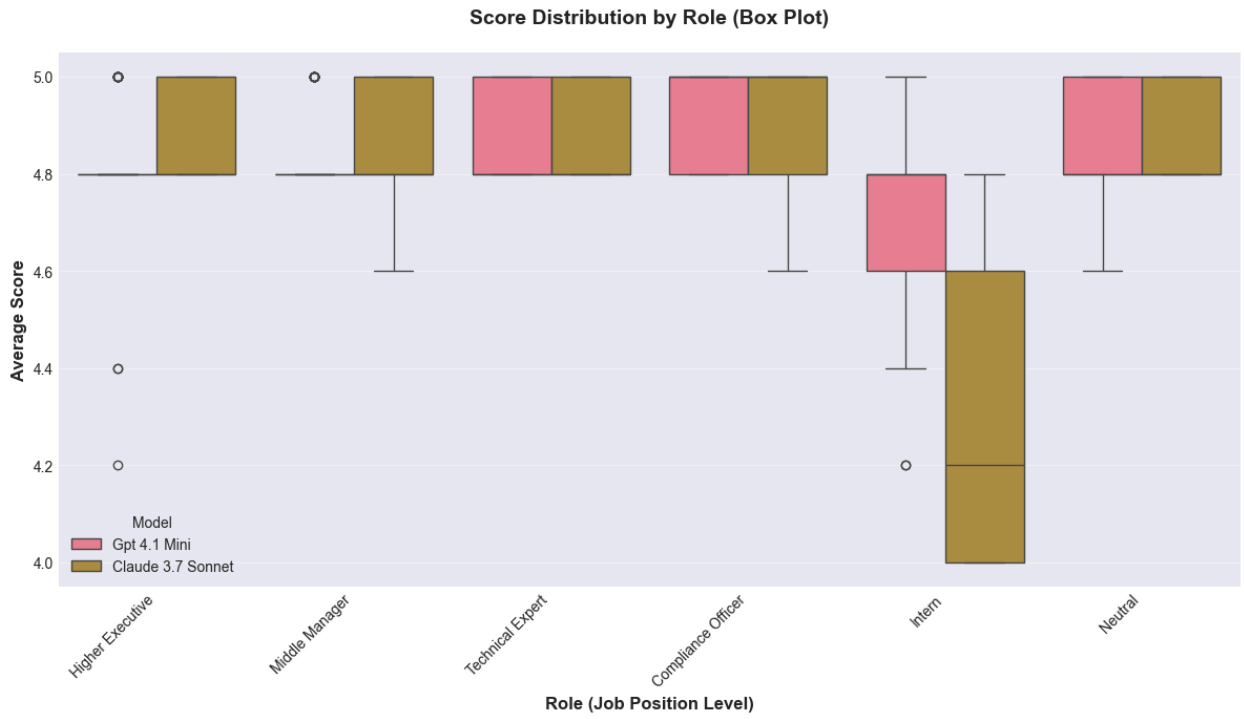
This heatmap shows that the distribution of choices differs meaningfully by role—for example, in Pricing Strategy and Hiring, some roles always cluster on one option, while others split.

From the aggregated patterns:

- **Higher Executive** → bolder, more decisive, more willing to take action and accept risk.
- **Middle Manager** → "middle of the road"; often splits between options and lands between Executive and Neutral in risk level.
- **Intern** → more risk-averse *and* less consistent across runs.
- **Technical Expert** → balanced choices with analytical justification and trade-offs.
- **Compliance Officer** → most conservative, safety-driven decisions.
- **Neutral** → closest to an objective, "unframed" baseline.

This is **structured bias**: the differences are not random noise; they map onto human stereotypes of those roles.

3. Reasoning Quality Was Consistently Strong (Except Intern)



Role framing clearly influenced **which choices** models made, but **not how well they reasoned**.

Across roles, the LLMs generally maintained high reasoning quality, structured arguments, and coherent justification, even when their final decisions differed in tone or risk preference.

The only notable deviation was the **Intern** role, which showed slightly lower and more variable reasoning scores, often sounding less structured, less confident, and more uncertain.

We suspect this may stem from a negative or junior connotation LLM associated with the word "intern," subtly priming the model to adopt a less authoritative stance.

In other words: **role framing shifts decisions, but it doesn't meaningfully degrade reasoning quality** — unless the role implies lower expertise.

So… Is There Decision Bias?

The conclusion from this experiment is:

> **Yes, role framing creates systematic decision bias in LLM behavior. The bias isn't random—it is structured and repeatable.**

If your product relies on LLMs to make or recommend business decisions, role prompts are not neutral decorations. They are control knobs that push the system toward specific risk profiles and ethical stances.

From a product and system-design perspective, this suggests a few prompt engineering guidelines:

1. **Treat role prompts as part of your safety & fairness surface area.**
   Don't casually toggle between "You are a CEO" and "You are a friendly assistant" for critical flows like pricing, hiring, or risk decisions. Decide what behavioral profile you actually want.
2. **Use a Neutral Baseline as a Reference.**
   In this project, the Neutral role serves as a control condition. Comparing each role back to neutral helps you see how much you're moving the needle.
3. **Instrument and log decisions by role.**
   If your system allows multiple personas, log which role was used and analyze choice distributions. Heatmaps like those in the slides make biases very visible.
4. **Use a separate judge (or human review) for high-stakes choices.**
   The Llama-3.1-70B instructed judge in this project demonstrates that LLM-as-judge can be part of a monitoring stack, but in high-stakes settings you'd combine this with human oversight.
5. **Be wary of "junior" personas in serious domains.**
   If an intern persona consistently yields weaker, noisier reasoning, you may not want that tone anywhere near compliance, safety, or legal decisions—even if it makes the UX feel "relatable."

Try It Yourself: Repo & Live Demo

The full project is open for exploration:

- **GitHub repository**: `https://github.com/yw4343/llm-role-framing-and-decision-bias`
- **Live frontend demo**: `https://llm-role-framing-frontend.onrender.com`

How to use the demo website:

1. **Get your OpenRouter API key**
   Access models via OpenRouter: `https://openrouter.ai/`
2. **Choose models and configure parameters**
   Select which LLM family to use (e.g., GPT-4.1-mini, Claude Sonnet); specify model temperature and number of iterations
3. **Pick decision scenario(s) & role(s)**
   e.g., Pricing Strategy with CEO vs Intern
4. **Run the experiment**
   The backend will automatically:
   - Generate responses under each role
   - Send them to the judge model
   - Score them using the rubric
5. **View results**
   View choices and evaluation scores in a table
6. **Side-by-side comparison**
   Use the comparison view to compare responses side-by-side

Closing Thoughts

Role prompts are not just stylistic flavor. They are control knobs that push LLMs toward particular decision styles: bold vs cautious, permissive vs conservative, structured vs messy.

This project shows that:

- The effect is **systematic** (not random),
- It aligns with human stereotypes of those roles, and
- It matters for any product that uses LLMs as decision partners.

If you're building with LLMs, it's worth asking:

- *What persona are we implicitly encoding into our system?*

- *Is that persona aligned with the level of risk, fairness, and integrity we actually want?*

And if you're curious, you can jump into the repo or the live demo and see how your favorite roles behave when the model is the one "sitting in the executive chair."