Randomness Injection

A jolt of creativity

- If you grow tired of tropes and clichés in your responses, I have an answer: Random Concept Injection
- This is something I do for parts of the system that do creative heavy-lifting. Oftentimes, AIs will hop to really tried and true answers to creative questions, which is great for reasoning well, but not so much for surprising an audience
- I use this to get names for characters that aren't baked into the training data, inject interesting concepts into simulations, and build out characters based on character archetypes
- It can be used for any list of random strings you'd like to be potentially incorporated into a particular decision-making process

Cost Considerations

Usage costs will likely increase

- Multilayered architectures cost more than single-prompt systems. Each layer is an API call, and those add up. If you're running a four-layer cycle on every user interaction, you’re potentially paying 4x what a single prompt would cost (depending on usage costs and tokens). That's the trade you're making for better results

- But they work better when properly configured. The question isn't "should I add more layers to save money”; it's "does the quality improvement justify the cost for my use case?" A customer service bot might not need four layers. A narrative engine generating premium content probably does

- Optimization strategies exist. Use cheaper models for simpler layers (correction doesn't need the most expensive model), cache aggressively for cyclical systems, and be honest about cutting layers that aren't pulling their weight. Every layer should earn its spot