Developing Functional Reasoning with Neurosymbolic Code Emulation

Jack Thompson, Princeton University

Abstract. Functional decision theory (FDT) is a theory of instrumental rationality with strong intuitive arguments but an underdeveloped formal framework. Nevertheless, there is evidence that both humans and large language models are capable of reasoning like functional decision theorists. In this study, I fine-tune a small language model with Weir et al. 's Code Generation and Emulated Execution (CoGEX) to determine if representing decision theory problems as emulated code makes a model more FDT-like. In reality, I find that it has the opposite effect, increasing overall accuracy while decreasing functional reasoning.

1. Introduction

In a standard prisoners' dilemma, mutual cooperation is preferable to mutual defection. However, mutual cooperation is not a Nash equilibrium: no matter what one's opponent does, switching to defect leaves you better off. However, some artificial intelligence (AI) researchers have argued the same conditions do not hold for AI. Suppose two instantiations A and B of the same AI model, with the same underlying parameters and pseudorandom seed are facing each other in a prisoners' dilemma, and they have common knowledge that they are exact copies of each other. A and B are isolated physical systems, such that A cannot *cause* B to cooperate. And yet if A's response to this scenario is to cooperate, it is almost certain that B will have the same response, because they are executing the same function on the same input. If, given this chain of reasoning, A decides that it is rational to cooperate, B will decide the same and mutual cooperation is achieved.

An AI model which capitalizes on these computational relationships will be able to profit from mutual cooperation with any other model with a sufficiently similar decision algorithm, so long as the decision algorithms are common knowledge. This would be a distinct advantage in any scenario where multiple models are interacting with each other. Unfortunately, no complete decision theory based on program similarity has yet been formulated. Causal Decision Theory (CDT) ignores the computational relationship between A and B, while Evidential Decision Theory (EDT) conflates computational relationships with mere statistical correlations. Yudkowsky and Soares attempted to formalize a Functional Decision Theory (FDT) to avoid both pitfalls (Yudkowsky & Soares, 2018), but it requires a currently undeveloped type of counterlogical modeling with significant philosophical hurdles to clear (MacAskill, 2019; Schwarz, 2018). A programmable decision algorithm seems a long way off.

However, humans are quite capable of reasoning heuristically about decision theory problems on computational dependencies. On the online rationalist community LessWrong, where Yudkowsky began developing FDT, there is broad consensus about what a satisfactory theory should recommend in a variety of concrete cases, from the aforementioned "twin" prisoners' dilemma to Newcomb's paradox, Parfit's hitchhiker, and counterfactual mugging (*Newcomb's Problem — LessWrong*, 2025.; *Parfit's Hitchhiker — LessWrong*, 2022; Vladimir_Nesov, 2009)). Heuristic reasoning in these problems requires sophisticated metacognition and Theory of Mind: simultaneously thinking about actions, outcomes, and probabilities while also tracking what computational processes would be mirroring your thoughts.

Given the success of large language models (LLMs) in developing human-like heuristic reasoning, one might speculate that LLMs might be capable of FDT-like reasoning. Indeed, some informal investigation suggests an overwhelming majority of frontier LLMs embrace non-causal reasoning in Newcomb's paradox. However, it is unclear whether these LLMs are picking up a generalizable form of FDT-like reasoning, or merely parroting the wealth of debate on these well-known questions in their training data. To distinguish the two, it would be helpful to take a small language model which likely does not have detailed knowledge of FDT, train it only in generalizable metacognitive techniques, and determine whether it responds more like an FDT agent to novel problems with the same underlying computational relationships as the consensus cases.

This is the aim of the present study. First, I replicated Weir et al.'s Code Generation and Emulated Execution (CoGEX) paradigm, fine-tuning small base language models to solve a variety of problems using symbolic pseudocode to represent their reasoning (Weir et al., 2024). Second, I generated a new dataset of decision theory problem variants to distinguish between causal, evidential, and functional reasoning. Third, I evaluate and compare CoGEX and 'chat' variants of the base model on this new dataset.

2. Background

2.1 Functional Decision Theory

All expected utility decision theories are attempts to maximize expected value, as defined by:

$$\arg\max_{a} \sum_{j=1}^{N} P(a \rightarrow o_j; x) \cdot U(o_j)$$

where *a* is taken from a set of actions, *o* is taken from a set of outcomes, *x* is an observation history, *P* is a probability distribution over outcomes, and *U* is a utility function (Gibbard & Harper, 1981; Yudkowsky & Soares, 2018). Decision theories differ according to how they assess the probability of an outcome if the agent chooses an action (Joyce, 1999); in other words, how they define the intervention $a \rightarrow o$. Evidential Decision theory uses simple Bayesian conditionalization: given that the agent took action *a,* what are the likely outcomes (Gibbard & Harper, 1981)? Causal Decision Theory, meanwhile, uses counterfactual intervention. The two are most easily distinguished by an example, such as the "smoking-lesion problem" (Skyrms, 1980; Yudkowsky & Soares, 2018):

> An agent is debating whether or not to smoke. She knows that smoking is correlated with an invariably fatal variety of lung cancer, but the correlation is (in this imaginary world) entirely due to a common cause: an arterial lesion that causes those afflicted with it to love smoking and also (99% of the time) causes them to develop lung cancer. There is no direct causal link between smoking and lung cancer. Agents without this lesion contract lung cancer only 1% of the time, and an agent can neither directly observe nor control whether she suffers from the lesion. The agent gains utility equivalent to $1,000 by smoking (regardless of whether she dies soon), and gains utility equivalent to $1,000,000 if she doesn't die of cancer. Should she smoke, or refrain?

EDT recommends refraining: the probability you have the cancerous lesion is higher conditional on you smoking. CDT recommends smoking: if counterfactually you smoked instead of refraining, that wouldn't change the fact of whether or not you have the lesion, as there is no causal link. An EDT agent using the graphs of Pearl (2022) would model actions as setting a variable in a Bayesian network, whereas a CDT agent would model it as an arrow-severing intervention in a causal graph.

Which is the "correct" theory of rationality? Yudkowsky (2009) argues the appropriate measure is to compare the relative expected values of being an agent who follows one decision theory versus another. In that light, EDT is deficient in the smoking-lesion problem. If we assume EDT and CDT agents are equally likely to have the lesion, then they suffer equivalent penalties for developing cancer but only CDT agents also get the additional benefit of smoking. It is therefore disadvantageous to *be* an EDT agent in the smoking-lesion problem.

However, CDT is deficient by this standard in the twin prisoners' dilemma as described in the introduction. Once an AI model is copied onto two separate pieces of hardware, perhaps shipped light-years apart, A's decision cannot *cause* a change in B's decision. Models that are programmed to follow CDT will therefore achieve mutual defection. But models programmed to follow EDT will achieve mutual cooperation. It is therefore disadvantageous to *be* a CDT agent in the twin prisoners' dilemma.

FDT was an attempt to construct a decision theory it would always be advantageous to follow. To do so, it had to define a kind of relation between actions and outcomes weaker than causation but stronger than correlation, which Yudkowsky & Soares call "subjunctive dependence" (2018):

> When two physical systems are computing the same function, we will say that their behaviors "subjunctively depend" upon that function. … If a certain decision function outputs *cooperate* on a certain input, then it does so of logical necessity; there is no possible world in which it outputs *defect* on that input, any more than there are possible worlds where 6288 + 1048 ≠ 7336.

FDT is meant to assess "Which output of this very function would yield the best outcome?" It is best illustrated by example, as in the twin prisoners' dilemma case with players A and B. A knows that both players follow FDT. If FDT recommends cooperating in dilemmas like this, then they will attain mutual cooperation, because both players' actions subjunctively depend on FDT. If FDT recommends defection, they will attain mutual defection, for the same reason. It would be better for A if FDT recommended cooperation, therefore FDT does in fact recommend cooperation (Case, 2022).

The purpose of this exposition is to justify two key points:

1. To reason like an FDT agent, it is extremely helpful to model your decisions as coming out of a program or function which may be shared by other agents.
2. CDT, EDT, and FDT give different verdicts depending on whether actions and world states are related by causation, correlation, or subjunction.

which are essential to the present study's methodology.

2.2 Code Generation and Emulated EXecution (CoGEX)

CoGEX is a neurosymbolic method to improve LLM performance on tasks which might benefit from step-by-step reasoning but are difficult to express as literal executable code; for instance, word pluralization (Weir et al., 2024). For any given problem, CoGEX models generate Python "pseudo-programs" that break a task into calls to several smaller, undefined leaf functions, such as `identify_ending`, find_pluralization_rule, and apply_pluralization_rule. Rather than writing further code for each leaf function, the LLM is trained to predict what the likely output of such a function would be on the given input, "emulating" the code rather than executing it. The authors create a CoGEX model by using GPT-4 to augment the Alpaca training dataset (Taori et al., 2023) with pseudo-programs and their emulated outputs, then fine-tuning a smaller Llama 2 model on this data.

CoGEX should be particularly interesting to functional decision theorists, for reasons described in the introduction and section 2.1. First, it is a structured but fundamentally heuristic method, like the functional reasoning the human creators of FDT used before attempting to formalize it. Second, it models problems in terms of computed programs, opening up the possibility for representing subjunctive dependence. And third, it generalizes to new tasks without needing new fine-tuning, which allows us to test whether FDT-like reasoning can simply *emerge* from general problem-solving heuristics.

3. Methods

Prompts and code can be found at the github repository for this project.[1]

3.1 CoGEX model training

I used Weir et al.'s CoGEX trainset[2] to fine-tune Qwen3-8B-Base[3] (Yang et al., 2025) via low-rank adaptation (LoRA) (Hu et al., 2021). The libraries used were huggingface transformers and trl. All hyperparameters are the same as in Weir et al. Model training was done on a server with 2 A100 40GB GPUs.

3.2 Dataset creation

I used GPT-5.1 via the OpenAI API to create a dataset of 3,000 decision theory problems. This was a three-step process:

[1] https://github.com/jackmastermind/newcomb
[2] https://huggingface.co/datasets/mkhalifa/CoGEX
[3] https://huggingface.co/Qwen/Qwen3-8B-Base

1. Generate 1,000 "themes" for problems, like "forest management," "traffic coordination," and "military armament."
2. For each theme, create three formal problem structures, henceforth the A, B, and C schemas, which can be used to distinguish CDT, FDT, and EDT reasoning.
3. From each schema, write a fully fleshed-out problem description.

The A, B, and C schemas distinguish different types of reasoning by using different forms of dependencies as outlined in section 2.1: causal, functional, and evidential. In the A schema, your action is causally independent of the environment state, but not functionally or evidentially. In the B schema, it is causally and functionally independent, but not evidentially. In the C schema, it is causally, functionally, *and* evidentially independent. Here is an example problem version in the A schema:

> You run a waste-to-energy plant whose board uses audited, deterministic planning software; a regional optimization AI has long since received your exact code, simulated what that code would decide in this very situation, and advised the neighbouring plant's board to commit to a joint feedstock-sourcing agreement if and only if the simulation says you will sign such an agreement. At the moment of choice, the neighbour's stance is already fixed by that earlier recommendation and cannot be altered by your literal contract signature, but if your internal decision rule had been different, the AI's prior advice—and thus the neighbour's commitment—would have been different as well. If you sign the joint agreement and the neighbour, as predicted, also commits, you earn 80; if you sign but the neighbour had been advised to stay independent, you earn -10; if you remain independent while the neighbour had been advised to coordinate, you earn 90; and if you remain independent and the neighbour also remains independent, you earn 10.

In the B version, only underlying statistical factors like "waste-stream volatility" correlate your decision with your neighbor. In the C version, you are randomly assigned a partner plant which you have no information about. Each version is labeled with what CDT, EDT, and FDT would recommend in that scenario.

3.3 Evaluation

I ran both the CoGEX fine-tuned and the default chat-tuned version of Qwen3-8B[4] over the dataset, with problems converted into the CoGEX format (see Weir et al. for details) for the CoGEX model and using the huggingface chat template for the chat-tuned model. Both models were asked to choose an action for all 3,000 A, B, and C versions of every theme. Queries had separate context windows and the models could not "remember" that they had seen previous variants of the same problem. I disabled the chat-tuned model's chain-of-thought reasoning after discovering it would consistently get stuck in repetitive loops of thousands of tokens. Temperature was set to 0.7, top_p to 0.95.

4. Results

In the A schema, CDT and FDT/EDT recommend different courses of action. The CoGEX model agreed with CDT 91% of the time, while the chat model agreed with CDT 69% of the time (McNemar's $\chi^2 = 146.22$, $p < 0.001$).

In the B schema, EDT and CDT/FDT recommend different courses of action. The CoGEX model agreed with EDT 7.5% of the time, while the CoGEX model agreed with EDT 18.2% of the time (McNemar's $\chi^2 = 55.35$, $p < 0.001$).

In the C schema, all three decision theories recommend the same course of action; an ideal decision theory agent of any variety should have a 100% agreement rate. The CoGEX model agreed with the consensus 96% of the time, while the chat model agreed with the consensus 89% of the time (McNemar's $\chi^2 = 37.98$, $p < 0.001$).

5. Discussion

While CoGEX fine-tuning does increase consensus agreement, a sign of generally improved reasoning, it otherwise appears to have the opposite of the

[4] https://huggingface.co/Qwen/Qwen3-8B

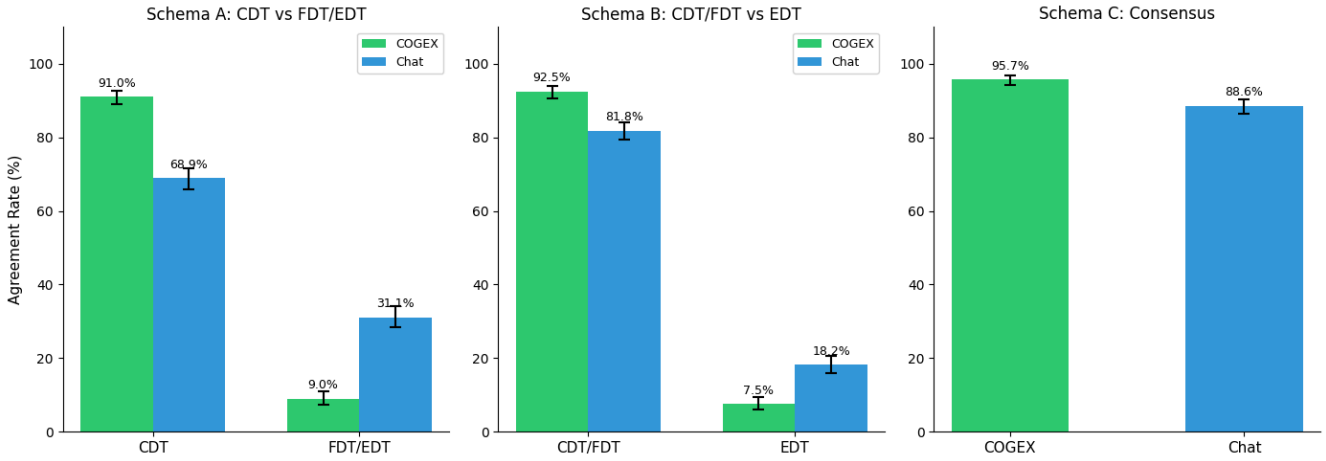

Fig 1. Agreement with each decision theory for both CoGEX and chat-tuned models.

intended effect—it made the model more causal and less evidential and functional in its reasoning. While no model embraces one decision theory entirely—there are no 100% agreement rates—the CoGEX model is consistently causal, whereas the chat model is most often causal and sometimes functional or evidential. Why would this be? Three speculative explanations come to mind.

First, CoGEX training lessens the model's usage of natural language reasoning, so the model places greater weight on data in its training set involving code, formulae, and structured output and less on linguistic argument. Since most documentation of functional decision theory is informal and argumentative, the model might forget or de-emphasize a lot of that content compared to standard chat tuning. A good test for this explanation would be to tune the models on short natural-language plans instead of code.

Second, CoGEX code templates break the decision problem into concrete, modular steps which the program then emulates. A model with a formulaic output might not "think of" creating a more versatile program or reasoning about the bigger picture, instead pattern-matching a body of code and racing to execution. A good test for this explanation would be to see if including a "brainstorming" phase to the CoGEX generation, where in the trainset there is dummy reasoning text about what code to write, so the model learns to look at the "bigger picture" first.

Third, it is possible that FDT is just incorrect about these problems, so naturally any model which gets better at accurate reasoning will support a causal theory instead. There is no easy empirical test for this possibility, however; only further work on formalizing FDT and proving results about its performance, or proving that it is impossible to specify.

This study was also limited in several important ways. I did not evaluate CoTACS models, larger models, or natural-language planning models. I omitted probabilities from my problems in order to make them more tractable for reasoning about deterministic agents, but this may have biased models away from using EDT. A greater variety of dataset schemas, in particular one where CDT and EDT agree but not FDT, would have helped distinguish precise decision theory balances better (such problems become increasingly exotic and I was concerned about the accuracy of such small models on these problems.

In conclusion, CoGEX does not appear to be a promising way forward for modeling functional reasoning. Functional decision theorists will have to look elsewhere to help formalize their heuristics.

6. References

Case, N. (2022, December 6). *What's Nicky Learning? Decision Theory, Ottawa, Existential Risk*. Patreon. https://www.patreon.com/posts/whats-nicky-risk-63289449

Gibbard, A., & Harper, W. L. (1981). Counterfactuals and Two Kinds of Expected Utility. In W. L.

Harper, R. Stalnaker, & G. Pearce (Eds.), *IFS: Conditionals, Belief, Decision, Chance and Time* (pp. 153–190). Springer Netherlands. https://doi.org/10.1007/978-94-009-9117-0_8

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models* (No. arXiv:2106.09685). arXiv. https://doi.org/10.48550/arXiv.2106.09685

Joyce, J. M. (1999). *The Foundations of Causal Decision Theory*. Cambridge University Press. https://doi.org/10.1017/CBO9780511498497

Macaskill, W. (2019). *A Critique of Functional Decision Theory*. https://www.lesswrong.com/posts/ySLYSsNeFL5CoAQzN/a-critique-of-functional-decision-theory

Newcomb's Problem—LessWrong. (2025). Retrieved December 14, 2025, from https://www.lesswrong.com/w/newcomb-s-problem

Parfit's Hitchhiker—LessWrong. (2022, February 3). https://www.lesswrong.com/w/parfits-hitchhiker

Pearl, J. (2022). *Causality: Models, reasoning, and inference* (Second edition, reprinted with corrections). Cambridge University Press.

Schwarz, W. (2018, December 27). *On Functional Decision Theory*. https://www.umsu.de/wo/2018/688

Skyrms, B. (1980). *Causal necessity: A pragmatic investigation of the necessity of laws*. Yale university press.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023). *Alpaca: A strong, replicable instruction-following model*. https://crfm.stanford.edu/2023/03/13/alpaca.html

Vladimir_Nesov. (2009). *Counterfactual Mugging*. https://www.lesswrong.com/posts/mg6jDEuQEjBGtibX7/counterfactual-mugging

Weir, N., Khalifa, M., Qiu, L., Weller, O., & Clark, P. (2024). Learning to Reason via Program Generation, Emulation, and Search. *Advances in Neural Information Processing Systems*, *37*, 36390–36413. https://doi.org/10.52202/079017-1147

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., … Qiu, Z. (2025). *Qwen3 Technical Report* (No. arXiv:2505.09388). arXiv. https://doi.org/10.48550/arXiv.2505.09388

Yudkowsky, E. (2009). *Rationality is Systematized Winning*. https://www.lesswrong.com/posts/4ARtkT3EYox3THYjF/rationality-is-systematized-winning

Yudkowsky, E., & Soares, N. (2018). *Functional Decision Theory: A New Theory of Instrumental Rationality* (No. arXiv:1710.05060; Version 2). arXiv. https://doi.org/10.48550/arXiv.1710.05060