
LUCR – Linking Utility and Compute Rate¹

Matt Duffy
Independent Researcher

With
Apart Research

Abstract

I developed LUCR (Local Utility–Compute Ratio), a framework for empirically measuring the gap between raw compute scaling and realized AI capabilities. Using data from Epoch AI (compute + capability indices) and METR (task completion benchmarks), I fit a hierarchical Bayesian model to link theoretical compute capacity to observed model performance.

My analysis quantifies the horsepower–capability delta, or the difference between what scaling laws predict, if we naively assume that cross-entropy loss directly translates to intelligence, and what models actually deliver. Algorithmic and translation efficiencies both act as a direct substitute for compute: a 20 % improvement in translation efficiency yields a similar median capability gain as doubling compute, algorithmic efficiencies gain even more.

This linkage is formalized in LUCR, which captures how capability changes per log-unit of compute and how that relationship shifts with architectural or training innovations. Across scenario analyses, algorithmic progress explains more near-term variance in capability growth than pure compute expansion, implying that compute, alone, may not translate to major advantages.

The LUCR framework provides an elasticity parameter that connects empirical benchmarking to compute-governance decisions and forecasting models, which enables forecasters to treat “algorithmic improvement” as a quantifiable substitute for compute rather than an unmodeled residual. This is a policy-relevant lever.

Keywords: compute scaling, algorithmic efficiency, LUCR, AI forecasting, capability elasticity

¹ Research conducted at the [The AI Forecasting Hackathon](#), 2025

1. Introduction

My central questions:

1. How can we measure the delta between raw compute horsepower (as predicted by Kaplan-style scaling laws (Kaplan, et al, 2020)) and realized model capabilities?
2. Given that linkage, how much algorithmic progress produces the same capability gain as one compute doubling?
3. Is the relationship between capability and compute—what we might refer to as the “horsepower-to-capability linkage”—tightening, loosening, or transforming as models evolve?

Forecasting AI timelines often relies on scaling laws that treat algorithmic improvement as an external constant. However, efficiency improvements, like architecture, training pipelines, and optimization, are endogenous and accelerating.

I hypothesize that the relationship between compute-driven and algorithmic-driven capability growth can be quantified directly from open benchmark data. To test this, I integrate Epoch’s compute–capability dataset (ECI) with METR’s task-performance results, defining a Local Utility–Compute Ratio (LUCR) as the derivative of METR task completion with respect to $\log_{10}(\text{compute})$.

This work contributes:

- A cross-dataset empirical model linking raw compute to real-world task capability.
- A policy-relevant elasticity parameter: algorithmic efficiency linked to equivalent compute.
- A Bayesian framework with uncertainty bands and scenario sliders for exploration.

2. Methods

Datasets:

- We use publicly available Epoch AI compute/capability tables and METR benchmark results as data sources.
- Epoch AI model table ($n = 80$ models) for training compute and capability index.
- METR benchmark results for 7 overlapping models (task completion 0–1).

Pipeline:

- Fit log-linear model: $ECI \sim \log_{10}(\text{Compute})$ $ECI \sim \log_{10}(\text{Compute})$ (Stage 1).
- Fit hierarchical Bayesian model: $METR \sim \alpha + \beta \times ECI$ (Stage 2).
- Define $LUCR = d(METR)/d(\log_{10} \text{Compute})$ as a joint slope.

Inference:

- MCMC (4 chains $\times$ 4 k draws) for posterior estimates.
- Model diagnostics: PPCs, BIC comparison, sensitivity analysis (GPT-2 exclusion).

Code is public on [GitHub](#).

3. Results

All referenced figures are in the Figures Appendix.

My analysis proceeds in four steps:

1. Visualizing the data and model fits,
2. Validating the model's robustness,
3. Analyzing the model's parameters (LUCR), and
4. Presenting the final scenario forecasts.

1. Data and Model Fits

Figure 1 presents the core data and model fits. The top-right panel (Stage 1) shows a clear log-linear relationship between compute and the Epoch Capabilities Index (ECI) for $n=80$ models. The bottom-left panel (Stage 2) shows the fit for the $n=7$ "overlap" models, bridging the ECI to my "subhuman" METR capability metric. The logistic fit captures the non-linear "S-curve" relationship as capability grows.

2. Model Validation and Sensitivity

To ensure my model is robust, I performed two checks. First, a posterior predictive check (Figure 2, left) shows my model's 90% and 50% highest-density intervals (HDIs) are well-calibrated and successfully capture the observed data points. Second, a sensitivity analysis (Figure 2, right) confirms that the "GPT-2 era" (pre-RLHF models) does not significantly bias the scaling slope (β). Excluding this era changes the slope by only $+3.1\%$ (or $+0.48\sigma$), justifying its inclusion in my final stratified model.

3. LUCR and Parameter Analysis

Figure 3 presents the 4-panel analysis of the model's core parameters. The key insight is the Local Utility–Compute Ratio (LUCR) (top-right), which measures $d(METR) / d(\log_{10} Compute)$. I find that LUCR is increasing in the current $10e^{24} - 10e^{28}$ FLOPs range, suggesting that for now, each 10x of compute yields more capability growth than the last. This is driven by the model's "algorithmic efficiency" (β), shown in the bottom-left panel. The uncertainty in this parameter is the primary driver of my forecast's variance.

4. Scenario Forecasts

Finally these forecasts use the model to forecast AGI timelines under 12 distinct scenarios. Figure 4 shows the median timeline for Near-AGI (0.8 METR, top panel) and AGI (0.9 METR, bottom panel) for all scenarios, color-coded by type. I

can see a vast range of outcomes, from 5.5 years in the "AGI Race + 20% Translation Efficiency" scenario to 20 years for compute restriction scenarios.

Figure 5 (left panel) highlights the probability of achieving these thresholds within 10 years. Under my Baseline (6-month compute doubling), the probability of reaching Near-AGI is only 4%. However, in an "Acceleration (AGI Race)" scenario, this probability jumps to 74%. Conversely, efficiency improvements alone (without compute acceleration) raise the probability to 34%. This shows that both compute scaling and algorithmic efficiency are critical, non-linear drivers of AGI emergence.

Scenario Explorations: The GitHub repo also contains an application, where users can input their own scenarios for compute doubling time, translation efficiency, and algorithmic efficiency, and see how the median timeline to AGI is impacted. An example image can be seen in Figure 6. Some definitions:

- *Algorithmic Efficiency:* Changes the rate at which compute improves intelligence
- *Translation Efficiency:* Changes the level at which improvement on all tasks translates to METR long-term tasks.

4. Discussion and Conclusion

Traditional scaling-law forecasts assume that capability improvements follow predictable loss reductions from increased FLOPs. My results suggest that this view is incomplete. The observed delta between theoretical scaling (Kaplan-style laws) and empirical capability (METR task completion) reveals a growing role for algorithmic efficiency—the ability of architectures, training regimes, and optimization methods to translate compute into useful cognition.

The LUCR metric quantifies this algorithmic improvement and translation efficiency. While LUCR shows an upward drift in the current compute regime (as seen in Figure 3), my model shows this is a local effect. Because LUCR is proportional to $METR * (1 - METR)$, it is mathematically guaranteed to peak and decline as capability saturates near 1.0. This implies that, without extending to superintelligence (ASI), pure compute scaling will face diminishing returns, making algorithmic efficiency the most critical factor for long-term capability gains.

Implications:

1. *Forecasting:* timelines could include a dynamic efficiency term; constant-scaling assumptions underestimate acceleration.
2. *Governance:* compute export controls may only delay capability milestones if algorithmic improvements outpace 15–20% per year.
3. *Research tracking:* LUCR can serve as a leading indicator of paradigm shifts by monitoring inflections in algorithmic and translation efficiency.

From a governance standpoint, LUCR provides a decision-relevant metric. If LUCR continues to rise, algorithmic innovation will outpace compute controls, suggesting that research transparency and coordination become the binding constraints.

Conversely, a declining LUCR would indicate diminishing returns to efficiency and renewed leverage for compute governance.

Limitations: current data are sparse, METR scores saturate near 1.0, and compute values for some models are estimated. Extending LUCR across richer benchmarks (ARC-AGI, HELM, BIG-Bench) would clarify whether efficiency gains remain exponential or plateau.

In short, LUCR reframes the question from “when will AGI arrive?” to “how efficiently are we converting hardware horsepower into capability?” The metric may better capture the true dynamics of AI progress.

5. References

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). *Scaling Laws for Neural Language Models*. arXiv preprint [arXiv:2001.08361](https://arxiv.org/abs/2001.08361).

6. Appendix

A. Figures

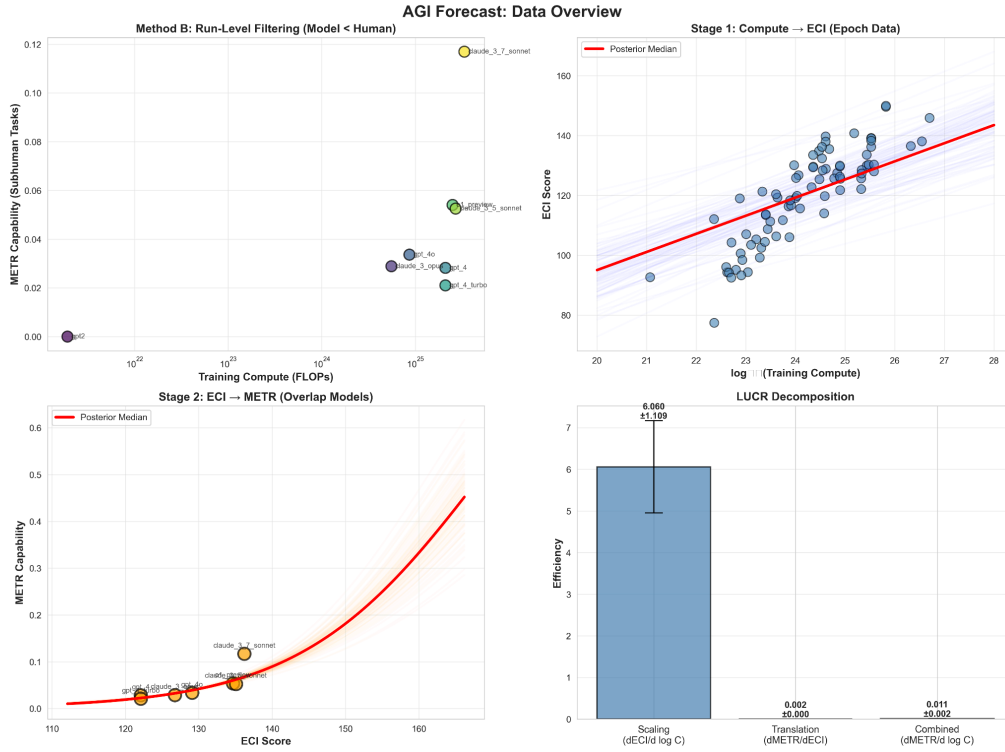


Figure 1 - Data Overview

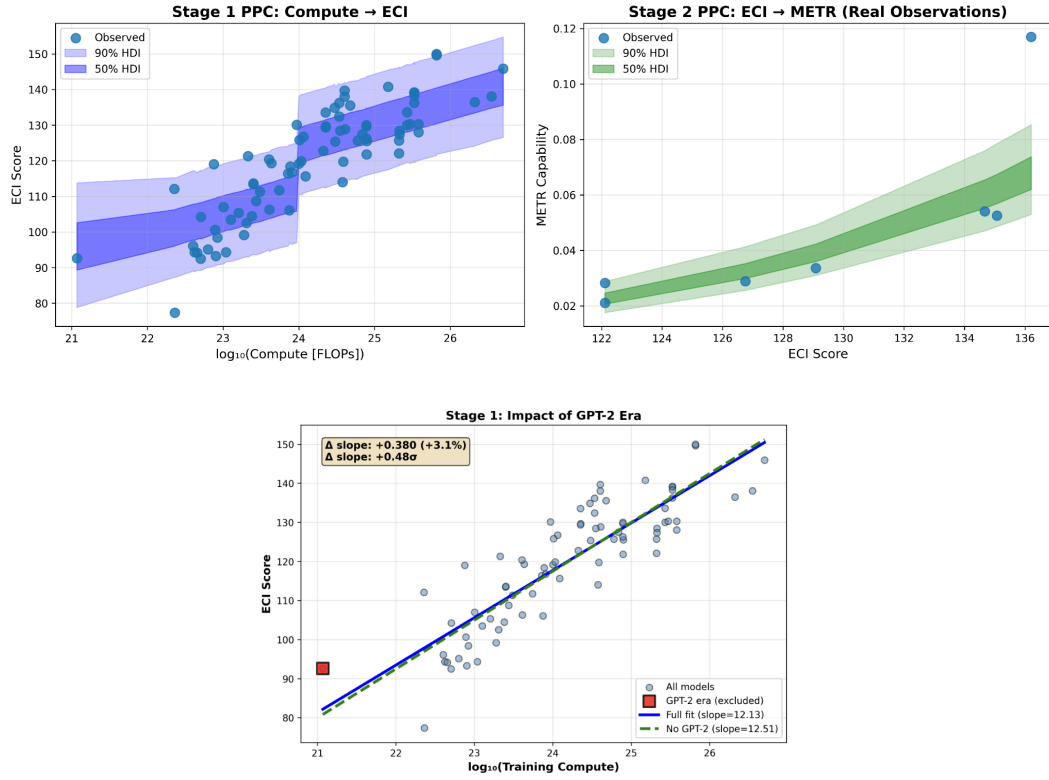


Figure 2 - Posterior Predictive Checks (top left and right) and GPT-2 Impact Analysis

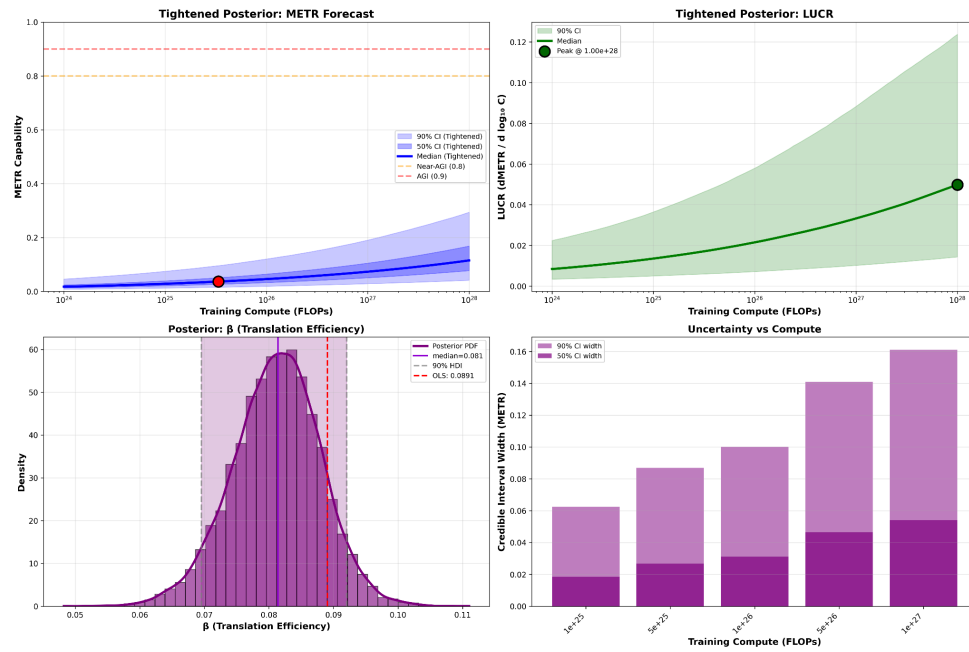


Figure 3 - LUCR Analysis

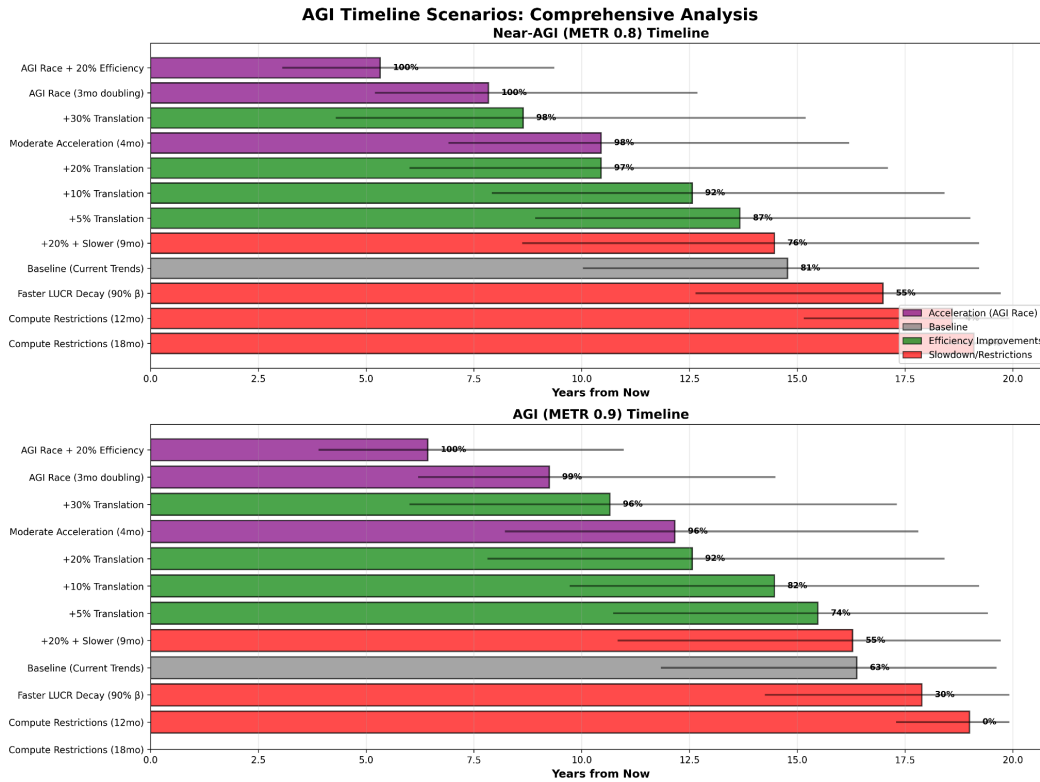


Figure 4 - Comprehensive Scenario Analysis

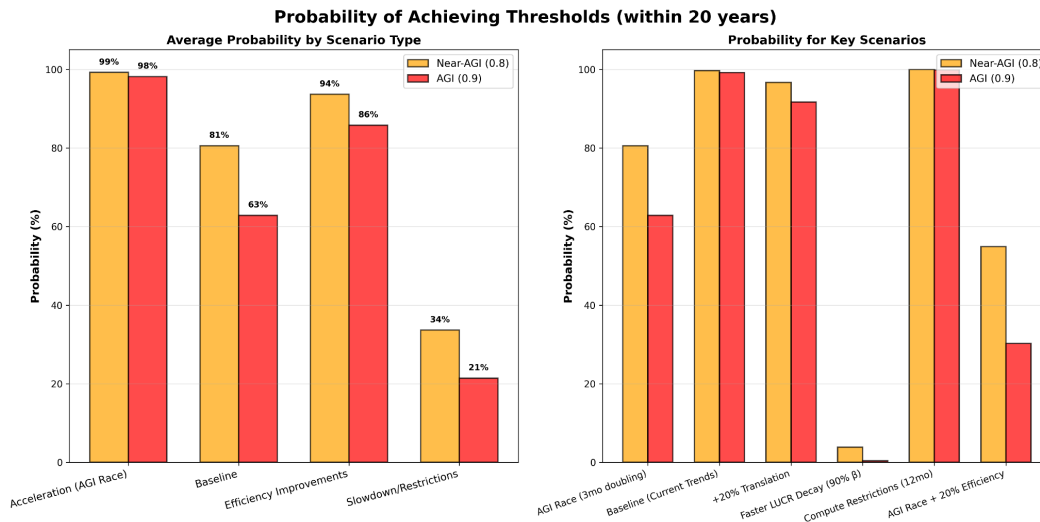


Figure 5 - Probability Analysis

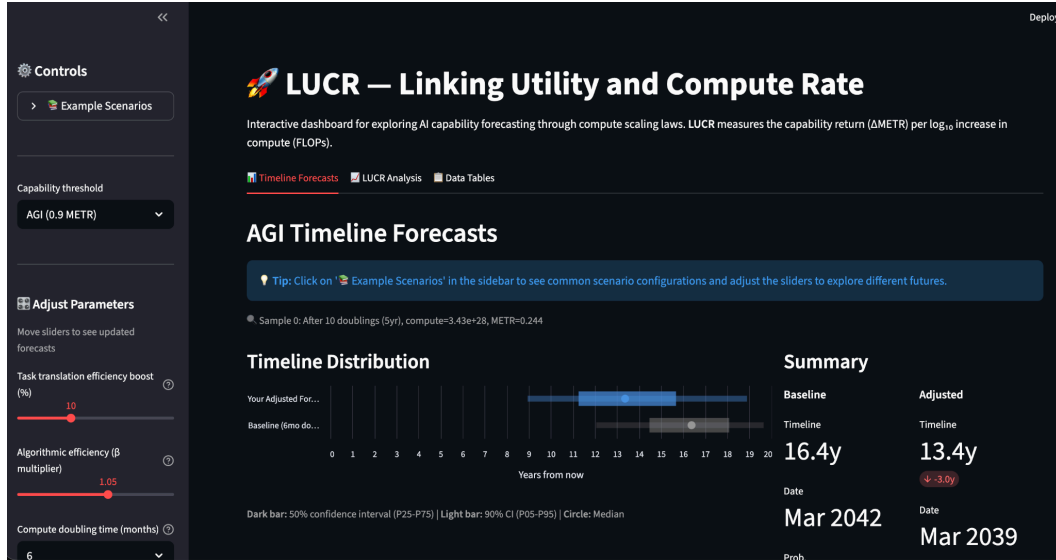


Figure 6: Application

B. Security Considerations

Forecasting capability emergence carries risk if misconstrued as predictive certainty. All parameters in LUCR are probabilistic and rely on open-source benchmarks. Future improvements should focus on interpretability, and more localized LUCR estimates rather than tighter AGI date estimates. I recommend publishing only aggregate metrics and maintaining transparency of methods to minimize misuse.