A Likelihood-Propagation Framework for Automatic, Sequential Inference

Alex Towell
atowell@siue.edu

May 19, 2025

Abstract

We formalize a *lkelihood propagation* approach that marries maximum likelihood estimation with the Fisher Information Matrix to deliver an automatic and highly efficient alternative to Bayesian sequential updating. Starting from one or more likelihood functions, the framework uses three primitives—likelihood, score, and information—to (i) obtain point estimates, (ii) propagate and combine uncertainty across time or data shards, and (iii) support decision-making and quantify predictive uncertainty, particularly in large-scale models, much like a posterior. We map the territory, articulate deep parallels with Bayesian conjugacy, and give precise algorithms, strengths, and weaknesses. While philosophically frequentist, the method is not *ad-hoc*: once the model is fixed, every update rule follows from likelihood theory and information geometry.

1 Introduction

Bayesian inference achieves coherent belief updates by multiplying prior and likelihood [1]. Yet in many engineering systems the computational burden of full posteriors (MCMC, variational inference) is prohibitive [2]. Practitioners therefore fall back on point estimation plus ad-hoc error bars. We show that a disciplined **MLE + FIM** pipeline gives:

- closed-form, *recursively* updatable estimates,
- principled uncertainty (via inverse information),
- algebraic composability for distributed or streaming data,

- identical asymptotic efficiency to Bayes with Jeffreys prior in regular models.

Despite decades of use in control (e.g., the Kalman filter [3]), signal processing (e.g., adaptive filters), and statistics (recursive estimation), the approach lacks a unifying name and exposition. We call it **Likelihood-Propagation Inference (LPI)**.

2 Preliminaries

Symbol	Meaning
$p(x \mid \theta)$	likelihood for datum x
$\ell(\theta) = \sum_i \log p(x_i \mid \theta)$	log-likelihood
$s(\theta) = \nabla_\theta \ell(\theta)$	score
$\mathcal{I}(\theta) = -\mathbb{E}[\nabla^2_\theta \ell(\theta)]$	(expected) FIM
$\hat{\mathcal{I}}(\theta) = -\nabla^2_\theta \ell(\theta)$	observed FIM

2.1 Point estimation

MLE $\hat{\theta}_n$ solves $s(\hat{\theta}_n) = 0$. In exponential families this is algebraic; otherwise use Newton or quasi-Newton iterations that *reuse* $\hat{\mathcal{I}}$.

2.2 Uncertainty via Information

To quantify the uncertainty associated with $\hat{\theta}_n$, LPI leverages Fisher Information, as described next. For regular models [4, 5]:

$$\sqrt{n}\,(\hat{\theta}_n - \theta_0) \xrightarrow{d} \mathcal{N}\big(0, \mathcal{I}^{-1}(\theta_0)\big). \tag{1}$$

Replacing θ_0 by $\hat{\theta}_n$ yields the familiar Wald (normal) confidence region. Crucially, **information adds** over IID samples:

$$\mathcal{I}_{1:n} = \sum_{i=1}^{n} \mathcal{I}_{x_i}. \tag{2}$$

2.3 Observed vs Expected FIM

The FIM can be formulated in two ways, observed or expected, with practical implications for its use. The *observed* matrix $\hat{\mathcal{I}}$ adapts to local curvature and generally outperforms the expectation for finite samples [6]; we treat it as default, noting both variants for completeness.

3 Likelihood-Propagation Inference (LPI)

The LPI framework operates on a specified collection of data-generating processes (DGPs),

$$\{p_k(x_k \mid \theta)\}_{k=1}^{K}, \tag{3}$$

which may be heterogeneous (e.g., a Gaussian sensor and a Bernoulli clickstream). A key property is that their log-likelihoods sum. Given this foundation, LPI defines how parameters are estimated and uncertainty is quantified by accumulating and combining likelihood-derived information through a few core operations, which we detail next.

Single-Batch Update Given batch B:

1. **Score accumulation**: $s_B(\theta) = \sum_{x \in B} \nabla_\theta \log p(x \mid \theta)$.
2. **Information accumulation**: $\hat{\mathcal{I}}_B(\theta) = -\sum_{x \in B} \nabla_\theta^2 \log p(x \mid \theta)$.
3. **Newton step**:
$$\theta^{\text{new}} = \theta^{\text{old}} - \hat{\mathcal{I}}_B^{-1} s_B. \tag{4}$$

Iterate until convergence (often one step suffices if batches are small).

Sequential / Streaming Update Maintain running $(\hat{\theta}_t,\ \hat{\mathcal{I}}_t)$. For new batch B_t:

$$\hat{\mathcal{I}}_t \leftarrow \hat{\mathcal{I}}_{t-1} + \hat{\mathcal{I}}_{B_t} \tag{5}$$

$$\hat{\theta}_t \leftarrow \hat{\mathcal{I}}_t^{-1}\big(\hat{\mathcal{I}}_{t-1}\hat{\theta}_{t-1} + \hat{\mathcal{I}}_{B_t}\hat{\theta}_{B_t}\big). \tag{6}$$

Exactly analogous to precision-weighted averaging of Gaussians.

Combining Independent Estimators If two estimators $(\theta_a, \mathcal{I}_a)$ and $(\theta_b, \mathcal{I}_b)$ arise from disjoint data, the optimal fusion is:

$$\theta_\star = (\mathcal{I}_a + \mathcal{I}_b)^{-1}(\mathcal{I}_a\theta_a + \mathcal{I}_b\theta_b) \tag{7}$$

$$\mathcal{I}_\star = \mathcal{I}_a + \mathcal{I}_b. \tag{8}$$

These operations form the core of LPI. While later sections offer deeper foundations, the fundamental concept is an automatic inference framework. Much like Bayesian methods, LPI provides a structured approach to learning from data. However, instead of propagating probability distributions (priors and posteriors), LPI propagates Fisher Information as the currency of uncertainty. The "automatic" nature stems from the fact that, once the likelihood model $L(\theta) = p(x|\theta)$ is specified, the rules for updating estimates and their associated information are mathematically determined, requiring minimal further decision-making.

4 Deep Parallels to Bayesian Inference

While LPI is philosophically frequentist, its operational structure and the role of information reveal deep parallels with Bayesian inference, particularly with conjugate Bayesian updating. These parallels highlight how LPI achieves similar inferential goals through a different theoretical lens, often with significant computational advantages. The core correspondences are summarized below:

- **Incorporation of Prior Knowledge vs. Initial State:** In Bayesian inference, prior beliefs about parameters are formally encoded in a prior distribution, $p(\theta)$. LPI, in its pure form, does not use subjective priors. However, the initial state of the aggregate estimate $(\hat{\theta}_0, \mathcal{I}_0)$ can be set using prior information, or regularization terms (Section 6) can act as pseudo-priors, with the Hessian of the regularizer contributing to the initial information matrix. This provides a mechanism, albeit different in interpretation, to incorporate pre-existing knowledge or to stabilize estimates in low-data regimes.

- **Data Assimilation:** Bayesian inference assimilates new data by multiplying the prior distribution with the likelihood function and then normalizing to obtain the posterior distribution, $p(\theta|x) \propto p(x|\theta)p(\theta)$. LPI, in contrast, assimilates data by *adding* the score (gradient of log-likelihood) and Fisher Information from the new data batch to the

existing aggregate quantities (Equations 5 and 6). This additive combination of information is algebraically simpler than the multiplicative and normalization steps in Bayesian updating.

- **Parameter Estimation (Central Tendency):** The Bayesian posterior mean, $\mathbb{E}[\theta|x]$, often serves as the Bayesian point estimate for θ. In LPI, the Maximum Likelihood Estimate, $\hat{\theta}$, which is the mode of the likelihood (and asymptotically the mode of the posterior under certain conditions), plays this role. LPI's sequential updates (Equation 6) show $\hat{\theta}_t$ as an information-weighted average of the previous estimate and the estimate from the new batch, akin to how posterior means are updated in Gaussian conjugate models.

- **Uncertainty Quantification (Dispersion):** Bayesian inference quantifies uncertainty about θ via the posterior covariance matrix, which is the inverse of the posterior precision matrix. In LPI, the Fisher Information Matrix (FIM), $\mathcal{I}(\theta)$, serves as the analogue of precision. Its inverse, $\mathcal{I}^{-1}(\theta)$, provides an (asymptotic) covariance matrix for the MLE $\hat{\theta}$, directly quantifying parameter uncertainty.

- **Sequential Updating and Conjugacy:** Bayesian conjugate updates offer closed-form solutions for the posterior when the prior and likelihood belong to compatible distributional families (e.g., Beta-Bernoulli, Normal-Normal). LPI achieves a similar operational simplicity through the additive nature of information (Equation 5 and 6). The updates for $\hat{\theta}_t$ and $\mathcal{I}_t$ are always closed-form (given batch estimates), regardless of the specific likelihood's family, assuming regularity conditions hold. This mirrors the computational ease of conjugate Bayesian models without being restricted to them.

- **Predictive Distributions:** To make predictions for new data x_{new}, Bayesian methods integrate over the posterior distribution of parameters: $p(x_{\text{new}}|x) = \int p(x_{\text{new}}|\theta)p(\theta|x)d\theta$. LPI typically uses a "plug-in" approach, $p(x_{\text{new}}|\hat{\theta})$, using the point estimate $\hat{\theta}$. However, as discussed in Section 9.3.1, parameter uncertainty from $\mathcal{I}^{-1}$ can be propagated via sampling or Laplace approximations [7] to generate richer predictive distributions that account for parameter uncertainty, thereby approaching the comprehensiveness of Bayesian predictive distributions.

- **Semantic Interpretation of Uncertainty:** A key philosophical difference lies in the interpretation of uncertainty. Bayesian posterior

probabilities represent degrees of *epistemic belief* about the parameters given the observed data and prior. The uncertainty quantified by LPI (e.g., confidence intervals derived from $\mathcal{I}^{-1}$) reflects *sampling variability*—how much the estimate $\hat{\theta}$ would vary if one were to repeat the data collection process under the same underlying true parameters θ_0.

The following table provides a concise summary of these parallels:

Concept	Bayesian	LPI (Frequentist)
Initial State	Prior $p(\theta)$	Initial $(\hat{\theta}_0, \mathcal{I}_0)$ / regularizer
Central Estimate	$\mathbb{E}[\theta \mid x]$	$\hat{\theta}_t \leftarrow \hat{\mathcal{I}}_t^{-1}\big(\hat{\mathcal{I}}_{t-1}\hat{\theta}_{t-1} + \hat{\mathcal{I}}_{B_t}\hat{\theta}_{B_t}\big)$
Uncertainty (Precision)	Posterior precision, e.g., $(\mathrm{Cov}(\theta\|x))^{-1}$	$\hat{\mathcal{I}}_t \leftarrow \hat{\mathcal{I}}_{t-1} + \hat{\mathcal{I}}_{B_t}$
Predictive Distribution	$\int p(x_{\text{new}}\|\theta)p(\theta\|x)d\theta$	Plug-in $\hat{\theta}_t$, optionally propagate $\mathcal{I}_t^{-1}$
Semantics of Uncertainty	Epistemic belief	Sampling variability

Table 1: Summary of Parallels between Bayesian Inference and LPI

Note: A particularly strong connection emerges when considering the Jeffreys prior, $p(\theta) \propto \sqrt{\det \mathcal{I}(\theta)}$ [8, 9]. With this non-informative prior, the Bayesian posterior mode and the inverse of the posterior curvature (as a measure of covariance) asymptotically match the MLE $\hat{\theta}$ and $\mathcal{I}^{-1}(\hat{\theta})$ from LPI. This reinforces the idea that LPI, while frequentist, often arrives at similar quantitative conclusions as a data-dominated Bayesian analysis, especially in large-sample regimes.

5 Strengths and Weaknesses

LPI offers several compelling advantages. Its computational lightness is a primary strength, as it avoids the high-dimensional integration often required by Bayesian methods. The framework is inherently online and distributed ready due to the additive nature of information, allowing for seamless updates with streaming data or across multiple processing units. Furthermore, LPI is model-agnostic, applicable to any model with a differentiable likelihood function. A significant practical benefit is its robustness to prior misspecification, simply because it does not require the specification of a prior distribution. From a theoretical standpoint, LPI aligns with

the natural-gradient view, meaning its update steps are invariant under reparameterization of the model [10].

Despite these strengths, LPI also has limitations. The uncertainty quantification it provides is *approximate*, relying on a normality assumption that may not always hold. There is no direct mechanism for incorporating subjective prior knowledge, which can be a drawback in scenarios where such information is valuable and available. Consequently, plug-in predictive distributions can under-represent tail risk compared to full Bayesian marginalization, which integrates over all possible parameter values. Finally, LPI can break down for non-regular models, such as those with mixture identifiability issues or parameters on the boundary of the parameter space, where the standard asymptotic properties of MLE and FIM do not apply.

6 Extensions and Variants

The core LPI framework can be extended and adapted in several ways to broaden its applicability and enhance its performance. One common extension is Regularization or the use of Pseudo-priors. By adding a penalty term $R(\theta)$ to the log-likelihood (e.g., L1 regularization [11] for sparsity, or L2 regularization [12] for shrinkage), its Hessian can be treated as an additional source of information, effectively guiding the estimate towards regions favored by the penalty. This can be particularly useful for ill-posed problems or to incorporate domain knowledge in a soft manner.

The choice between Observed versus Expected FIM offers another axis of variation. While the observed FIM, calculated from the second derivatives of the actual log-likelihood at the MLE, is often adopted by default due to its superior performance with finite samples [6], the expected FIM can be more stable and is particularly useful for theoretical analyses and design calculations, such as determining optimal experimental designs.

For non-stationary data streams where the underlying data-generating process may change over time, Forgetting Factors can be introduced. This typically involves scaling the accumulated information $\mathcal{I}$ by a factor $\rho < 1$ at each step, thereby down-weighting older data and allowing the estimates to adapt more quickly to recent changes.

LPI naturally connects to Stochastic Natural Gradient Descent (SNGD) [13]. In SNGD, mini-batch scores (gradients) are pre-conditioned by a running estimate of the Fisher Information Matrix. This can be seen as an application of LPI principles to large-scale optimization, where the FIM guides the parameter updates along the path of steepest ascent in the Riemannian

manifold defined by the information geometry.

Finally, for situations where the Gaussian approximation of uncertainty is insufficient but full Bayesian integration is too costly, Hybrid Laplace-Propagation methods can be employed. These approaches retain the core Gaussian approximation for the bulk of the parameter distribution but incorporate third-order (or higher) derivative corrections, such as those from Laplace approximations [7], to better capture skewness or other non-Gaussian features in the posterior or predictive distributions.

7 Information-Theoretic Foundations

While LPI is presented through the lens of statistical inference, its deepest connections lie in information theory [14, 15], potentially revealing more fundamental principles than the probabilistic framing alone.

7.1 From Shannon to Fisher: The Information Pipeline

The Fisher Information Matrix quantifies the *curvature* of the log-likelihood surface [16], but it also represents the rate at which a parameterized distribution carries information about its parameters. This dual interpretation connects LPI to fundamental information-theoretic principles:

- **Information flow**: Data reduces uncertainty about parameters by conveying information, quantified precisely by the FIM
- **Maximum entropy inference**: LPI preserves exactly the information present in the data—no more, no less
- **Efficient coding**: Parameter values maximizing likelihood correspond to optimal coding of data under the model

Shannon's entropy of data $H(X) = -\mathbb{E}[\log p(X)]$ and the Fisher information about parameters connect through the fundamental equation:

$$\mathcal{I}(\theta) = \mathbb{E}_\theta \left[\left(\frac{\partial}{\partial \theta} \log p(X|\theta) \right)^2 \right] = \mathrm{Var}_\theta \left[\frac{\partial}{\partial \theta} \log p(X|\theta) \right] \tag{9}$$

This reveals Fisher information as the variance of the score—how sensitively the log-likelihood responds to parameter changes—providing a direct link to entropy's focus on surprise and information content.

7.2 Entropy Minimization and LPI

Maximum likelihood estimation minimizes Kullback-Leibler divergence between the empirical distribution $\hat{p}_{\text{data}}$ and model distribution p_θ:

$$D_{\text{KL}}(\hat{p}_{\text{data}} \| p_\theta) = \mathbb{E}_{\hat{p}}\left[\log \frac{\hat{p}(x)}{p_\theta(x)}\right] = -H(\hat{p}) - \mathbb{E}_{\hat{p}}[\log p_\theta(x)] \tag{10}$$

Since the entropy of the empirical distribution $H(\hat{p})$ is constant with respect to θ, MLE equivalently maximizes $\mathbb{E}_{\hat{p}}[\log p_\theta(x)]$, precisely the expected log-likelihood under the data distribution.

LPI's parameter updates can therefore be reinterpreted as incremental entropy minimization steps. Each batch B reduces the cross-entropy between the observed data distribution and the model, with the Fisher information guiding the optimal step size in information space rather than parameter space [17].

7.3 The 20 Questions Principle

Consider the classic game of 20 questions, where we identify an object through yes/no questions. Information theory tells us that with n binary questions, we can uniquely identify one of 2^n objects—if each question optimally splits the remaining uncertainty.

LPI embodies this principle: each data batch provides information that optimally reduces uncertainty about parameters. The Fisher information precisely quantifies how much uncertainty a new observation can eliminate, guiding:

- Which parameters learn fastest (higher Fisher information components)
- How to design optimal experiments (maximize expected information gain)
- When we've accumulated sufficient data (information thresholds)

7.4 Mutual Information and Learning Dynamics

As data accumulates, the mutual information between parameters θ and observed data X increases, bounded by the entropy of the parameter distribution. LPI's sequential updates track this information acquisition process, with:

$$I(\theta; X) \approx \frac{1}{2} \log \det(\mathcal{I}) + \text{constant} \tag{11}$$

revealing how the determinant of Fisher information relates to mutual information. This connection explains why LPI's uncertainty estimates (via $\mathcal{I}^{-1}$) align with asymptotic posterior uncertainties—both reflect the same fundamental information-theoretic quantities.

8 The Algorithm

While simple in its conceptual form, LPI's implementation benefits from careful algorithmic design. The following pseudocode provides a complete implementation template that emphasizes the incremental nature of information accumulation:

Algorithm 1 Likelihood Propagation Inference (LPI) - Core Algorithm

Initialize: θ_0, $\mathcal{I}_0 \leftarrow$ regularization matrix or 0
for $t = 1, 2, \ldots$ **do**
    Observe data batch B_t
    Compute batch MLE: $\hat{\theta}_{B_t} \leftarrow \arg\max_\theta \ell_{B_t}(\theta)$
    Compute batch FIM: $\mathcal{I}_{B_t} \leftarrow -\nabla^2 \ell_{B_t}(\hat{\theta}_{B_t})$
    Update Information: $\mathcal{I}_t \leftarrow \mathcal{I}_{t-1} + \mathcal{I}_{B_t}$
    Update Estimate: $\theta_t \leftarrow \mathcal{I}_t^{-1}(\mathcal{I}_{t-1}\theta_{t-1} + \mathcal{I}_{B_t}\hat{\theta}_{B_t})$
    Optional: Apply forgetting factor $\mathcal{I}_t \leftarrow \rho\mathcal{I}_t$ for non-stationary data
end for
Return: $\theta_t, \mathcal{I}_t$ (estimate and uncertainty)

For large models where computing or inverting the full FIM is impractical, structured approximations [18] maintain the essence of information propagation while scaling to millions of parameters:

- **Diagonal FIM**: $\mathcal{I} \approx \text{diag}(I_{11}, I_{22}, \ldots)$ for parameter-wise adaptive steps (related to methods like AdaGrad/RMSProp [19]).
- **Block-diagonal**: Captures parameter group correlations while maintaining tractability.
- **Kronecker-factored**: $\mathcal{I} \approx A \otimes B$ for neural network layers [18].
- **Low-rank + diagonal**: $\mathcal{I} \approx UV^T + D$ balances expressivity and computational efficiency.

These approximations preserve the information-additive nature of LPI while making computation feasible.

8.1 Integration with Existing Systems

The beauty of LPI lies in how it naturally embeds into existing ML pipelines:

1. **Replace optimizers**: LPI can be implemented as an optimizer that maintains and leverages curvature information.
2. **Enhance existing optimizers**: Information accumulation can augment SGD or Adam [19] with principled adaptation.
3. **Enable distributed training**: Information additivity allows parameter servers to properly combine estimates from workers [20].
4. **Improve transfer learning**: Information matrices provide natural weighting when transferring knowledge between domains.

Information-theoretic metrics derived from the FIM also enable principled early stopping, learning rate schedules, and architecture search based on the rate of information accumulation.

9 Modern Computational Perspectives

The fundamental LPI primitives—likelihood, score, and information—scale elegantly to modern machine learning systems through several key technological developments.

9.1 Automatic Differentiation and Computational Graphs

Modern automatic differentiation (AD) frameworks [21] have revolutionized the practical application of LPI by making its core operations virtually automatic:

- **Score computation**: $s(\theta)$ becomes a simple backpropagation call.
- **Information matrix**: $\mathcal{I}(\theta)$ computation through forward or reverse-mode differentiation.
- **Parameter updates**: Natural gradient steps via efficient matrix operations.

This democratizes LPI for practitioners who can now focus on model specification rather than derivative computation. The additive nature of information in LPI aligns perfectly with modern computational paradigms for distributed and federated learning.

For high-dimensional models where full FIM computation is prohibitive, AD frameworks enable efficient approximations that preserve LPI's core computational advantages:

- Diagonal approximations that only store parameter variances.
- Block-diagonal approximations that capture parameter group correlations.
- Kronecker-factored approximations for neural network layers.
- Low-rank plus diagonal structures that balance expressivity and efficiency.

These approximations extend LPI's applicability to models with millions or billions of parameters while maintaining its fundamental algebraic elegance.

9.2 Optimization Landscapes and LPI in Deep Learning

Finding $\hat{\theta}_n$ that solves $s(\hat{\theta}_n) = 0$ faces new challenges in deep learning contexts where likelihood landscapes are no longer well-behaved [22]. The LPI framework naturally accommodates advanced optimization strategies:

- **Stochastic gradient ascent**: Mini-batch score accumulation (Section 3.1) becomes stochastic gradient estimation.
- **Momentum methods**: Compatible with score-based updates while preserving information geometry [23].
- **Adaptive learning rates**: Can be viewed as diagonal approximations to the FIM, aligning with LPI's information-weighting principle.
- **Natural gradients**: Direct implementation of the core LPI update equation $\theta^{\text{new}} = \theta^{\text{old}} - \mathcal{I}^{-1}s$.
- **Distributed optimization**: Leverages the additive property of scores and information across data shards.

The recursive update formulas of Section 3.3 provide a principled foundation for these methods, offering theoretical grounding for many heuristic practices in deep learning optimization.

9.3 LPI and Foundation Models

Large Language Models (e.g., GPT [24], BERT [25]) and other foundation models represent perhaps the most ambitious application of likelihood-based estimation to date. Despite their scale and complexity, these systems remain fundamentally likelihood-driven.

In the context of such high-dimensional models, the traditional inferential goal of interpreting individual parameters becomes less relevant. Instead, the primary focus shifts to understanding and quantifying the uncertainty in the model's *predictions*—such as the distribution over the next token in LLMs. The parameter uncertainty captured by the FIM (and its approximations) serves as a crucial intermediate step to derive these predictive uncertainties. For example, by sampling parameters $\theta^{(s)}$ from their approximate distribution $\mathcal{N}(\hat{\theta}, \hat{\mathcal{I}}^{-1})$, one can generate an ensemble of output distributions, enabling the construction of confidence intervals for top-k predictions or other hypothesis testing procedures related to model outputs.

9.3.1 Deriving and Utilizing Predictive Uncertainty in LLM Outputs

The LPI framework's ability to quantify parameter uncertainty via $\hat{\theta}$ and $\hat{\mathcal{I}}^{-1}$ offers a direct pathway to richer predictive uncertainty for LLM outputs, particularly for the next-token distribution. This goes beyond simple point predictions and can inform more nuanced generation strategies.

Constructing Confidence Intervals for Next-Token Probabilities: Given the LPI-derived parameter estimate $\hat{\theta}$ and its approximate covariance $\hat{\mathcal{I}}^{-1}$, we can characterize the uncertainty in the predicted next-token probabilities as follows:

1. **Parameter Sampling:** Draw S samples of the parameter vector, $\theta^{(s)} \sim \mathcal{N}(\hat{\theta}, \hat{\mathcal{I}}^{-1})$, for $s = 1, \ldots, S$. This step leverages the asymptotic normality of the MLE, where $\hat{\mathcal{I}}^{-1}$ is the estimated variance.

2. **Ensemble of Predictive Distributions:** For a given input context, and for each sampled parameter vector $\theta^{(s)}$, compute the full probability distribution over the vocabulary V for the next token: $P^{(s)} = \{p(v_j|\text{context}, \theta^{(s)}) \text{ for all } v_j \in V\}$. This results in an ensemble of S predictive distributions.

3. **Token-Specific Probability Intervals:** For any specific token v_j in the vocabulary (particularly for those tokens that are candidates

under standard decoding, e.g., the top-k tokens according to the mean prediction $p(v_j|\text{context}, \hat{\theta})$), we now have a collection of S probability values: $\{p(v_j|\text{context}, \theta^{(1)}), \ldots, p(v_j|\text{context}, \theta^{(s)})\}$.

4. **Confidence Interval (CI) Estimation:** From this collection of S probabilities for token v_j, a $(1-\alpha)\times 100\%$ confidence interval, $[L_j, U_j]$, can be estimated. A straightforward method is to use the empirical percentiles of the sampled probabilities (e.g., the $\alpha/2$ and $1-\alpha/2$ percentiles).

This procedure yields not just a point probability for each potential next token, but also a range reflecting the model's uncertainty about that probability due to parameter uncertainty.

Leveraging Predictive CIs in LLM Decoding Strategies: These token-specific CIs can directly inform and enhance common LLM decoding strategies:

- **Uncertainty-Aware Top-k/Top-p Sampling:** Standard top-k or top-p (nucleus) sampling [26] typically relies on point estimates of token probabilities. LPI-derived CIs allow for more sophisticated selection:
  - *Robust Selection:* The sampling pool could be restricted to tokens whose *lower confidence bound* L_j exceeds a certain threshold, or tokens could be ranked by L_j. This prioritizes tokens that are reliably probable, potentially reducing the chance of nonsensical or low-quality continuations.
  - *Exploratory Selection:* Conversely, tokens could be considered if their *upper confidence bound* U_j is high, even if their mean probability $p(v_j|\text{context}, \hat{\theta})$ is not in the initial top set. This encourages exploration of tokens the model is uncertain about but considers plausible under some parameter configurations, potentially leading to more diverse or creative outputs.
  - *Adaptive Nucleus:* The size of the nucleus p in top-p sampling could be dynamically adjusted based on the aggregate uncertainty (e.g., average width of CIs for high-probability tokens). Higher uncertainty might warrant a larger nucleus for more exploration.
- **Quantifying Output Reliability:** The width of the CIs $(U_j - L_j)$ for chosen tokens can serve as a direct measure of the model's confidence

in its own output probabilities, useful for downstream tasks or for signaling when human review might be necessary.

By incorporating these LPI-derived predictive uncertainty measures, LLM generation can move beyond simple likelihood maximization towards more controllable, robust, or diverse text generation, directly reflecting the information (and its limitations) captured by the model parameters.

Key aspects of LPI are particularly salient for these models:

- Training objectives are variations of log-likelihood maximization, directly connecting to LPI's first primitive.
- Parameter estimation uncertainty (via the FIM), even if not used for direct inference on individual parameters, provides valuable signals. These include guiding active learning [27] and exploration, informing principled early stopping criteria based on information gain (e.g., when $\log\det(\mathcal{I})$ plateaus), or refining learning rate schedules.
- Information additivity enables principled distributed training and continual learning [28, 29]. Similarly, LPI provides a robust framework for fine-tuning pre-trained foundation models. In this scenario, the parameters of the pre-trained model ($\hat{\theta}_{\text{pre}}$) and its associated Fisher Information matrix ($\mathcal{I}_{\text{pre}}$, possibly approximated) serve as a powerful, data-derived pseudo-prior. Initializing the LPI updates with $(\hat{\theta}_{\text{pre}}, \mathcal{I}_{\text{pre}})$ means that new parameters are learned by balancing the likelihood from the fine-tuning data against a quadratic penalty for deviating from $\hat{\theta}_{\text{pre}}$. This penalty, $\frac{1}{2}(\theta - \hat{\theta}_{\text{pre}})^T\mathcal{I}_{\text{pre}}(\theta - \hat{\theta}_{\text{pre}})$, effectively acts as a soft constraint. Such an approach is closely related to minimizing a divergence (e.g., a second-order approximation to the KL divergence between a Gaussian centered at θ and one at $\hat{\theta}_{\text{pre}}$ with precision $\mathcal{I}_{\text{pre}}$) from the "distribution" embodied by the pre-trained model. This allows for the preservation of general "common sense" knowledge captured during pre-training while adapting the model to new, task-specific data using $\mathcal{I}_{\text{fine-tune}}$.
- Regularization techniques map naturally to LPI extensions described in Section 6.

The success of these systems demonstrates that even as models become increasingly complex, the core principles of LPI—maximum likelihood estimation guided by information geometry—remain foundational. Indeed, many challenges in modern AI (catastrophic forgetting, efficient fine-

tuning, uncertainty calibration [30]) can be reframed and potentially addressed through the lens of information propagation.

10 Illustrative Example: Deep Learning Model Training

Consider training a deep neural network (DNN) for classification using a cross-entropy loss, which is equivalent to maximizing the log-likelihood of a categorical distribution. LPI provides a lens to understand and enhance this process:

- **Stochastic Updates as LPI Steps**: Training with mini-batches can be viewed as a sequence of LPI updates. For each mini-batch B_t:
  1. The gradient of the loss $\nabla_\theta \mathcal{L}_{B_t}$ is the negative score $-s_{B_t}(\theta)$.
  2. The (approximate) Fisher Information Matrix $\mathcal{I}_{B_t}$ can be estimated (e.g., using empirical FIM, diagonal approximations like in Adam/RMSProp, or Kronecker-factored approximations).
  3. An optimizer step, especially one like natural gradient descent, takes the form $\theta_t \leftarrow \theta_{t-1} - \eta \mathcal{I}_{B_t}^{-1} s_{B_t}(\theta_{t-1})$, directly analogous to the LPI update, or more generally, $\theta_t \leftarrow \mathcal{I}_t^{-1}(\mathcal{I}_{t-1}\theta_{t-1} + \mathcal{I}_{B_t}\hat{\theta}_{B_t})$ if we consider $\hat{\theta}_{B_t}$ as the conceptual MLE for that batch.
- **Information Accumulation and Regularization**: The total information $\mathcal{I}_t = \sum_k \mathcal{I}_{B_k}$ (or a running average) reflects the model's accumulated knowledge. Techniques like Elastic Weight Consolidation (EWC) [28] for continual learning explicitly use the FIM to penalize changes to parameters important for previous tasks, which is a direct application of LPI's information-weighting principle.
- **Uncertainty and Model Analysis**: Approximations to the FIM provide insights into parameter uncertainty, which, while not typically used for interpreting individual parameters in large DNNs, are instrumental for deriving *predictive uncertainty* for model outputs (e.g., class probabilities or next-token distributions). The inverse FIM, $\mathcal{I}_t^{-1}$, offers a principled (though approximate) covariance matrix for $\hat{\theta}_t$, forming the basis for sampling parameters to estimate the variability of predictions. Furthermore, FIM-derived metrics can identify parameter sensitivities, guide pruning or quantization, and inform training dynamics like early stopping based on information saturation.

While full FIM computation is often intractable for large DNNs, the LPI framework motivates and provides theoretical grounding for many successful heuristics and approximations used in modern deep learning, framing them as attempts to efficiently propagate likelihood-derived information.

11 Discussion & Future Work

Unifying terminology: we advocate **Likelihood-Propagation Inference (LPI)** to emphasize (i) likelihood as the sole probabilistic object, and (ii) additive propagation of information. Future lines include exploring LPI for non-regular models, which are common in scenarios with latent variables or boundary conditions where standard FIM properties may not hold; developing methods for automatic curvature corrections, potentially beyond the Gaussian approximation inherent in FIM, to better handle complex likelihood landscapes; and integrating robustness techniques (e.g., sandwich covariance estimators [31, 32]) to ensure reliable uncertainty quantification even under potential model misspecification.

12 Conclusion

LPI offers a rigorously grounded, markedly more computationally efficient, and practically powerful alternative to full Bayesian updating—recovering similar algebraic structure without priors or costly integration. For many real-time and large-scale applications, it delivers the "automatic" inference pipeline practitioners crave, while retaining a clear, principled link to classical likelihood theory. By focusing on the direct propagation and accumulation of Fisher Information, LPI offers a distinct yet powerful paradigm for understanding learning and uncertainty, rooted directly in information theory, providing an alternative to the Bayesian tracking of full posterior distributions over parameters.

References

[1] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3rd edition, 2013.

[2] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[4] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer, 2nd edition, 1998.

[5] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[6] B. Efron and D. V. Hinkley. Assessing the accuracy of the maximum likelihood estimator: Observed versus expected fisher information. *Biometrika*, 65(3):457–487, 1978.

[7] L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

[8] H. Jeffreys. *Theory of Probability*. Oxford University Press, 1939.

[9] C. P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2nd edition, 2007.

[10] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[11] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[12] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[13] R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. arXiv preprint arXiv:1301.3584, 2013.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[15] S. Kullback. *Information Theory and Statistics*. Wiley, 1959.

[16] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

[17] S. Amari. *Information Geometry and Its Applications*. Springer, 2016.

[18] J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *PMLR*, pages 1107–1115, 2015.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[20] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. A. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25*, pages 1223–1231, 2012.

[21] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

[22] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 27*, pages 2933–2941, 2014.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *PMLR*, pages 1139–1147, 2013.

[24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. OpenAI Blog, 2018.

[25] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[26] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. Often available as arXiv 2019.

[27] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[28] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[29] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[30] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pages 1321–1330, 2017.

[31] H. White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.

[32] P. J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 221–233, 1967.