The Road from LLMs to LCMs

(Semantic Mismatch and Uncertainty)

Prof. Dr. Petre Dini, ex-AT&T/Cisco Systems, Inc., USA (retired)
Prof. Ing. Luigi Lavazza, Università degli Studi dell'Insubria, Italy

Themes

- LargeLMs - SmallLMs - TinyLMs
- DeepSeek and Optimization
- Visual LLMs
- Hallucinations
- LLMs and LCMs
- Concepts and LCMs Core Layer
- (Semantic) Mismatching
- Uncertainty

Size and Optimization

- **Size of languages models**
  - Large, small, tiny
  - Software, embedded (SoC)
- **Computation complexity**
  - Trillion of parameters
  - Repetitive computation
- **Optimisation**
  - Finding the best patterns
  - Accept adapted accuracy
  - Clustering
  - Drop extra computation cycles after a satisfactory outcome

- **Examples**
  - Lenses/dioptre
  - Syslog messages
- **Not all applications/services need exceptional accuracy**
  - Classification of needs
  - Clustering of exploration space
    - not all data have value
    - not everything deserves extra-computation power
    - powerful kids-like language models

LargeLMs - SmallLMs - TinyLMs

Model Size (>1 billion parameters, 100 million - 1 billion parameters <100 million parameters)

Training Data (size and context-based datasets; unbalanced input)

Computation Needs (hardware, energy; tiny: run on edge devices/phones)

Latency (model complexity; tiny: suitable for real-time applications)

Use Cases (complex tasks, specialized tasks; tiny: lightweight applications, IoT devices, mobile apps)

Cost (high due to compute and storage requirements; moderate; minimal resources)

Accessibility (accessible via cloud APIs, easily deployable; tiny: embedded, minimal overhead)

DeepSeek

- **DeepSeek's AI assistant surpassed OpenAI's ChatGPT in the Apple App Store**
- **DeepSeek: DeepSeek-R1**: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning
- arXiv:2501.12948v1 [cs.CL] 22 Jan 2025

- The essence: instead of individual assessment, make an estimate based on group assessment; in larger groups, a lot of computation resources is saved.

- In fact, **the surprise** is that the results are not affected (to feel it, see below that "ε" threshold), by this apparent simplification; that is, comparatively, the individual diopters are in reality like 2.2499999999978999999..., but the eye glasses are of 2, 2.25, 2.50; you can see relatively well with 2 and 2.25, it depends on the distance, the light, the size of the text, the state of fatigue, the color of the ink, the shape of the letters, etc... (in AI, in many applications, there is no need of many decimals); of course, it depends on the field.

Group Optimisation

2.2.1. *Reinforcement Learning Algorithm*

Group Relative Policy Optimization In order to save the training costs of RL, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which foregoes the critic model that is typically the same size as the policy model, and estimates the baseline from group scores instead. Specifically, for each question q, GRPO samples a group of outputs $\{o_1, o_2, \cdots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_θ by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \quad (1)$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (2)$$

where ε and β are hyper-parameters, and A_i is the advantage, computed using a group of rewards $\{r_1, r_2, \ldots, r_G\}$ corresponding to the outputs within each group:

$$A_i = \frac{r_i - mean(\{r_1, r_2, \cdots, r_G\})}{std(\{r_1, r_2, \cdots, r_G\})}. \quad (3)$$

DeepSeek Metric

- **Model Complexity** (small to large): Complex neural networks, object recognition and scene semantics, would align more closely with the complexity of small to large language models. These models are sophisticated enough to manage significant data inputs and provide detailed, context-aware outputs.
- **Computation Needs** (less than the largest language models used for generative text). Tasks are complex, yet more focused on specific visual data processing, which can sometimes be optimized more efficiently than broad, multi-faceted language understanding.
- **Use Cases** (for applications requiring advanced visual comprehension). Mostly autonomous vehicles, advanced security systems, or augmented reality environments (in depth and learning capabilities that mirror the strengths of larger language models in handling nuanced and varied data).
- **Resource Optimization** (typical of tinyLMs); Mobile devices or edge computing scenarios. However, the base technology would still lean towards larger, more resource-intensive setups.

DeepSeek vs Visual LLMs

Feature DeepSeek / Visual LLMs

Core Function: Object localization and scene understanding / Understanding and generating content from visual data

Data Handling: Optimized for specific visual tasks (clustering and approximation) / Wide range of visual data, with large and diverse datasets for training

Model Complexity: Moderate to high f(implementation), advanced algorithms for efficiency and speed / High; complex architectures, large-scale visual and textual data

Accuracy: Computation needs: Less accuracy for speed and computational efficiency via clustering / For high accuracy, extensive data and computational resources to minimize error

Computation needs: Optimized vs general visual LLMs; real-time applications / Significant computational power, specialized hardware like GPUs or TPUs

Use Cases: Real-time applications: surveillance, autonomous vehicles, and robotics (quick decision-making) / Broad: image captioning, object detection, and complex scene analysis.

Adaptability: Fine-tuned for specific environments or operational conditions: efficiency and performance / Highly adaptable to new tasks through transfer learning and fine-tuning; for visual understanding

Themes

Hallucinations (on purpose - they are human - vs induced - by the model -)

LLM Resource Consumptions (precision vs tailored precision)

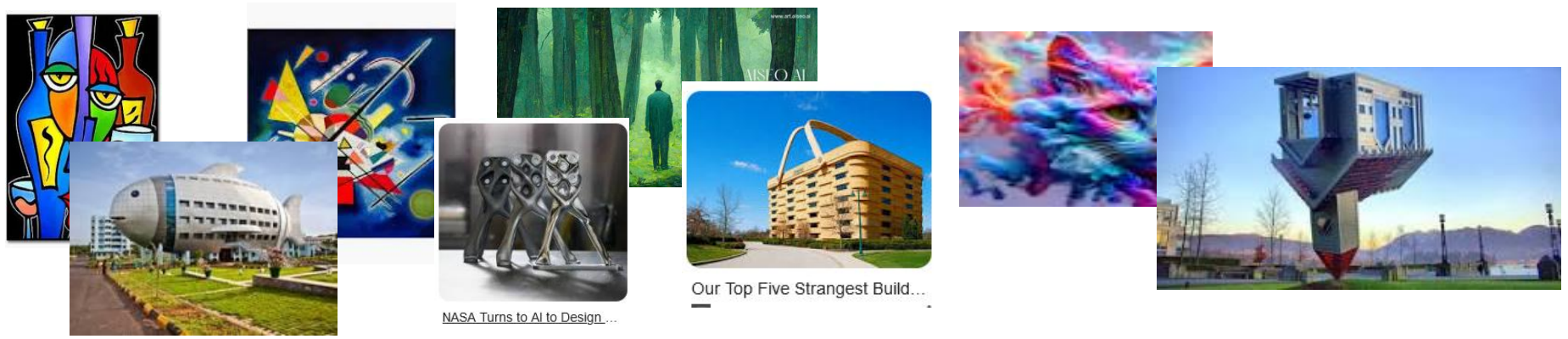
Semantic Mismatching (dentification & disambiguation)

LLMs, SLMs, tinyLMs (resource vs accuracy)

LLMs vs LCMs (asset management)

Uncertainty Estimation of LLMs (entropy, fuzzy/weights)

Hallucinations



2+2 = 5, all lines are straight, 2 = 10, log(-3) = x
Horses ride a bicycle over the shining cloud of dust!!
Aliens will invest in The House of the rising Sun!
Alice's Adventures in the Wonderland

Metaverse
VR AVR
Immersion

John published a paper on Mars in the year 2500 [24].
There are 5 Planets and one Galaxy only.
Army of X plans to attack Y country at midnight, 2024, June 30.

Recommended references on a geology paper.
[x] Pierre, Title 1
[z] Jacobs, Title 2
[y] Stan, Title 3
Note: [x] doesn't exist
[z] has another title
[y] is a carpenter instructions book

LLMs-based: -
very good summarization of information they are fed with, even only less than 1% validated as true
- very good mixed (4-5-6 …) language**s**, correct punctuation, correct grammar, spelling correction on context-based intuition
- helpful at the informative level, like white papers, very quickly obtained and quite comprehensive
- assumes user's familiarity and experience with a given domain; see, selection an oscilloscope for 5G spectrum, financial aspects,…

Resource Optimization

Analogy: Syslog messages, Eye diopter, Nature Shapes Brain Captures,… (*context-dependent*)

(running for 100% is context-dependent, resource optimization, including water & energy)

LLMs: Stable/adopted vocabulary

(synonyms, idioms, …)

(intensive process, repetitive, hardly reproducible, trillions of parameters, overfitting, underfitting, input balancing, hallucinations, …)

DeepSeek: clustering (optimization), rounding meaning (similarity, sufficient approximation)

LCMs: ad hoc concepts, open range, uncontrolled definitions, concept management, definition conflicts, evolution control

LLMs, LCMs and Mismatches

xLMs: tailored selection

Optimization
Rounding meaning (similarity, sufficient approximation)

LCMs
Concepts repositories
Concepts ownership

Domain-oriented tools selection
Accuracy-by-request
Precision-by-request
Authenticity-by-request

Mismatch detection

Usually, post effect

Hardly detectable until damage is done

Disambiguation
Uncertainty
Human-in-the-loop
Extra-resources

License responsibility
Guarantees
Penalties

Outsourced disambiguation and reconciliation operations

Semantic Mismatch (i)

One LLM: : Polysemy, Pragmatism, Temporal aspects, Contextual roles, Discourse contradiction

> User input: "What is the best bank in the city?"

> LLM response: "The best river bank is the one with the nicest walking path..."

> 👉 Mismatch: Interpreted bank as a geographical feature, not a financial institution.

> User input: "Who painted the 'Mona Lisa'?"

> Follow-up: "What else did she do?"

> LLM response: Lists activities of the woman in the portrait

➢ 👉 Mismatch: Treated "she" as the subject of the painting, not maintaining focus on Da Vinci.

> Prompt: "Explain why nuclear power is clean energy."

> Later in same response: "Nuclear waste is a major environmental pollutant."

> 👉 Mismatch: Coherence failure — two partially correct facts, but semantically contradictory when juxtaposed.

Disambiguation and Reconciliation Strategies

A. Prompt Engineering Techniques
B. Tool-Assisted Disambiguation
C. Conversation Memory and Role Anchoring
D. Model Fine-Tuning or Re-ranking
Train or fine-tune on domain-specific corpora with contextually grounded labels
Use re-ranking models to pick the best disambiguated answer from top-k completions
E. Semantic Consistency Checkers
Deploy post-hoc semantic contradiction detectors
Ensure no mutually exclusive claims in the same output

Semantic Mismatch (ii)

Between LLMs: Divergent Definitions, Concept Boundaries, Conflicting Scope, Context Assumptions, Domain-Specific Vocabulary, Inconsistent Entity Linking, Cultural or Regional Bias

- **Prompt:** *"Define what counts as a 'mild adverse reaction' in vaccine trials."*
- **LLM-A (e.g., GPT-4):**
  "A mild adverse reaction includes local pain, slight fever, or fatigue that resolves within 24–48 hours."
- **LLM-B (e.g., Claude or LLaMA):**
  "Mild reactions refer to non-serious symptoms that do not require medical intervention."
- 👉 **Mismatch:** Different **granularity** and **focus** — one uses clinical timing, the other legal/medical threshold.
- **Prompt:** *"What does 'freedom of speech' mean in the context of digital platforms?"*
- **LLM-A:**
  "It is the right of users to express opinions freely, limited by platform guidelines."
- **LLM-B:**
  "It is the protection from government censorship; platforms may set their own rules."
- 👉 **Mismatch:** Different **legal-cultural frames** — U.S. constitutional vs platform governance views.

Disambiguation and Reconciliation Strategies

A Prompt Standardization

Use controlled, context-rich prompts:

Add role (e.g., "as defined in EU regulation X")

Specify domain scope (e.g., "medical context", "U.S. law")

B Comparative Output Analysis

Run semantic similarity metrics (e.g., cosine similarity, BERTScore) on outputs

Use human-in-the-loop review or majority consensus for high-stakes answers

C Meta-Evaluation with Third Model

Use a neutral LLM to assess

D Ontology or Domain-Specific Schema Anchoring

Bind both LLMs to a shared controlled vocabulary, ontology, or taxonomy (e.g., SNOMED CT, WordNet, LexInfo)

E Trust Layer or Explanation Generator

Ask each LLM to explain:

"Why did you define this term this way? What sources or assumptions are you using?"

LLMs and LCMs

Here's the big picture:

sql Copy Edit

```
User ↔ LLM ↔ LCM ↔ External Knowledge / Tools
```

- **LLM** handles: natural language understanding, interaction, generalization
- **LCM** handles: structured concept grounding, validation, abstraction, explanation
- **Goal**: use LCM to "filter", validate, or enrich the LLM's decisions/actions

Concept Mapping Layer (Bridge)

Converts LLM-parsed input into conceptual tokens using:

- Schema mapping (e.g., concepts from ConceptNet, DBpedia)
- Ontology alignment (e.g., OWL-based matching)
- Embedding clustering or symbolic tagging

text Copy Edit

```
"Explain gravity on the Moon."
↓
["Gravity", "Moon", "Mass", "Acceleration"]
```

LCM Core Layer

This is the reasoning brain composed of:

Knowledge Graph: concepts + relations

Reasoning Engine: for inference (e.g., description logic, rule engines)

Conceptual Simulation: mechanisms for analogical or causal reasoning

Could be built with:

Component	Technologies
> Ontologies	OWL, Protégé
> Graph DBs	Neo4j, RDF/SPARQL
> Reasoning	Pellet, HermiT, Rule Engines
> Learning	Meta-Concept Embeddings (Vec2Graph)
> Inference	CLIPS, MiniKanren, ASP

LLMs vs LCMs

Feature / Aspect	LLM (Large Language Model)	LCM (Large Concept Model)
🔍 Primary Objective	Model linguistic patterns and generate coherent text	Model, abstract, and reason with concepts and relationships
🧱 Core Representation	Tokens, word embeddings, attention distributions	Concepts, relations, hierarchies, logical rules
📚 Training Data	Text corpora (web, books, conversations, code)	Ontologies, knowledge graphs, structured data, task schemas
🧠 Learning Style	Pattern learning via self-supervised next-token predictions	Symbolic abstraction, graph expansion, property induction
🧩 Compositionality	Implicit, via transformer layers and positional embeddings	Explicit, via concept composition (e.g., part-of, is-a, causes)
🔄 Reasoning	Mostly statistical (e.g., analogies, surface inferences)	Symbolic or hybrid reasoning (deductive, abductive, causal)
🧠 Contextuality	Strong in recent context windows (e.g., 128k tokens)	Strong in structured semantic context and background knowledge
🧪 Interpretability	Often opaque (“black-box”)	More explainable (conceptual graphs, rule sets, ontologies)
🤖 Use Cases	Text generation, dialogue, summarization, translation	Planning, abstraction, categorization, analogical problem solving
🔌 Tool Integration	Prompt-based, external tools via plugins	Programmatic interfaces, rule engines, structured APIs
🧠 Generalization	Strong zero-shot/few-shot generalization in language tasks	Strong generalization in conceptual domains, especially with schema
🛠️ Limitations	Hallucinations, shallow semantics, weak long-term abstraction	Brittle logic, data sparsity, poor natural language fluency

Uncertainty (i)

Themis AI - CSAIL Alliances - MIT
https://cap.csail.mit.edu

- **Types of Uncertainty**

  a. *Aleatoric Uncertainty* (data-based)
  Comes from inherent noise or ambiguity in the input (e.g., multiple valid interpretations).
  b. *Epistemic Uncertainty* (model-based)
  Stems from a lack of knowledge or data; reducible with more/better training data.

- ***For LLMs Specifically***

  Prompting Variability: Asking the same question in different forms and comparing the answers can reveal uncertainty.
  Chain-of-Thought Diversity: Sampling multiple reasoning chains and analyzing their consistency.

Uncertainty (ii)

Common Uncertainty Estimation Techniques

1. Log Probability / Token-Level Entropy

The model's log-probability distribution over tokens can be used to estimate uncertainty.
Higher entropy = more uncertainty.
Useful in: Language modeling, Autocomplete confidence, Chain-of-Thought token path variability

2. Monte Carlo Dropout

Apply dropout during inference (not just training) multiple times.
Analyze the variance in outputs to estimate uncertainty.
Pros: Simple, applicable to Transformer layers.
Cons: Adds computational cost.

3. Ensemble Methods

Use multiple LLMs trained differently or with different seeds.
Diverse outputs or high disagreement → higher uncertainty.
Especially used in high-stakes applications (e.g., clinical or legal generation).

4. Temperature Scaling (in softmax)

Tuning the softmax temperature changes output confidence.
High temperature → more uniform predictions → higher estimated uncertainty.

5. Bayesian Approaches

Treat model parameters as probability distributions (e.g., Bayesian Transformers).
Often too heavy for production-scale LLMs but used in research.

6. Conformal Prediction

A statistical framework providing prediction intervals or sets.
Can be adapted for NLP tasks, like classification or span extraction.

7. Calibration Metrics

Measures how well the model's confidence corresponds to reality.
Common ones:
- Expected Calibration Error (ECE)
- Brier Score
- Negative Log Likelihood (NLL)

Uncertainty Estimation (example)

Medical Decision-Making Support

- **Goal**: Use an LLM (like Med-PaLM or GPT-based tools) to assist with clinical reasoning, diagnosis, or summarizing reports, while estimating confidence in its recommendations.
- 🔍 **Technique: Token-Level Entropy + Ensemble Sampling**
- **Setup:**
- Task: Answer clinical questions based on a patient case or radiology report.
- Model: An LLM fine-tuned on biomedical corpora.
- Approach:
  - Sample **multiple answers** using different temperatures or prompt phrasings.
  - Measure **token-level entropy** and **output diversity**.

Example:

Prompt: “What is the most likely diagnosis for a 65-year-old patient with chest pain, elevated troponin, and ST elevation in leads II, III, aVF?”

LLM answers:

Myocardial infarction (common)
Inferior STEMI (more specific)
Acute coronary syndrome (more general)

Uncertainty Estimation:

Token entropy for each word is low for “myocardial infarction” → high confidence.
When more diverse responses appear (e.g., pulmonary embolism, angina), entropy increases → signal to escalate to human.

Metric Used:

Entropy (H) of output tokens

$H=-\sum_{i=1}^{n} p_i \log p_i$
$H=-\sum_{i=1}^{n} p_i \log p_i$

(where pi are token probabilities)

Uncertainty Estimation (example)

Still subjective

Entropy – extra computation resources

Weights / ~ fuzzy logic

Probabilities

Token-Level Entropy

For the example medical phrase completion:

Tokens: "myocardial infarction", "angina", "pulmonary embolism"

Assigned probabilities: [0.75, 0.15, 0.10]

Computed Entropy:
$H = -\sum p_i \log_2 p_i \approx 1.05$ bits

→ This is low entropy, indicating high confidence in the top token ("myocardial infarction").