Do LLMs Think Ahead? A Literature Review and a Minimal Plan for Multi-Token Prediction

Harley T. Gribble

Table of contents

Abstract

This paper reviews recent evidence on whether decoder-only transformers trained with next-token prediction implicitly "think ahead," and surveys architectural and training modifications that make look-ahead behavior explicit via **multi-token prediction**. We focus on two anchor works: (1) Wu et al. (2024), who analyze **myopia** using linear probes and correlation analyses on hidden states, and (2) Meta AI (2024), who train **multiple output heads** to predict tokens (t+1,…,t+K) from the final hidden state and pair this with **self-speculative decoding** to accelerate inference. We conclude with a minimal experimental plan: start from a **pretrained LM**, attach **two auxiliary heads**, **train those heads only** (no trunk updates), and evaluate whether independent heads can extract reliable multi-step predictions before attempting full multi-head joint training.

1. Introduction

Modern LLMs are trained with a next-token objective, which appears inherently myopic. Yet practitioners observe that models often maintain longer-range coherence and planning. Two complementary research directions explore this tension:

- **Diagnostics**: Are future-token signals already present in hidden states?
- **Mechanisms**: Can we train models to **explicitly** predict multiple future tokens efficiently and accurately?

This review synthesizes recent results along both axes and motivates a minimal, low-risk experimental plan that builds atop an existing pretrained model.

2. Background (concise)

- **Decoder-only transformer**: Input tokens (→) embeddings (→) stack of pre-LN blocks (LayerNorm (→) masked self-attention (→) residual (→) LayerNorm (→) FFN (→) residual) (→) final hidden state $h_T^{(L)}$ (→) output linear layer ("unembedding") (→) logits (→) softmax.
- **Training objective**: Cross-entropy at each timestep between predicted distribution and the true next token.
- **Probing**: Freeze the LM; fit a simple model (often linear) from hidden states to a target (e.g., a future token or a synthetic label) to test **linear accessibility** of information.
- **Speculative/self-speculative decoding**: Use a fast/draft path to propose multiple next tokens, then verify with the base model in fewer passes than vanilla autoregressive decoding.

3. Related Work

3.1 Wu et al. (2024): *Do Language Models Think Ahead?* (arXiv:2404.00859)

Goal. Test whether hidden states encode information about **future** tokens beyond (t+1), and whether models behave **myopically** (only optimizing for immediate next-token prediction).

Key ideas. - Construct or analyze **myopic** settings (e.g., restrict/alter gradient flow so updates don't rely on distant future positions). - On **frozen LMs**, train **linear probes** on hidden states $h_t^{(\ell)}$ to predict future-dependent targets. - Run **per-neuron correlation**

between each hidden dimension and the future target as a sanity check (if single coordinates align strongly, future info may be trivially present).

Findings (high-level). - Hidden states can contain signals predictive of future tokens; in small models this may be *incidental* (features helpful for (t+1) also correlate with (t+k)); larger models show **clearer, more explicit** future-relevant features. - Myopia is not absolute: even when trained myopically, some future information appears **linearly decodable** at intermediate layers.

3.2 Meta AI (2024): *Scaling Multi-Token Prediction with LLMs* (arXiv:2404.19737)

Goal. Improve efficiency and potentially quality by training models to **predict multiple future tokens** from the same final hidden state.

Architecture. - Add **K parallel output heads**: for the final state $h_T^{(L)}$, each head (k) produces logits for token (t+k).
- **Training loss**: sum (or average) the cross-entropy across heads. To save memory, compute **head losses sequentially**, backpropagating and **accumulating gradients** in the shared trunk.

Inference (self-speculative decoding). - **Draft** (K) tokens via multi-token heads (one pass).
- **Verify** all (K) tokens with the standard next-token head (one pass over the extended sequence).
- Accept matching prefix; repeat. This can cut required passes dramatically when drafts are accurate.

Findings (high-level). - **Efficiency gains** are robust; **accuracy** improvements are task-dependent (strong for code; mixed for general language at large scale).
- Multi-token training may reduce “derailments” by encouraging better choices at branching points, but optimal training recipes are still active research.

4. Concepts and Terms (reader-oriented)

- **Myopia**: Optimizing strictly for (t+1) can under-reward representations useful for (t+2,…).

- **Linear probe**: A frozen-LM diagnostic; if a linear model can predict a target from $h_t^{(\ell)}$, then that information is **linearly accessible** at that point.

- **Neuron–target correlation**: Per-dimension Pearson correlation between a hidden coordinate and a target; a coarse, sanity-check signal.

- **Multi-token heads**: Additional linear projections $z^{(k)} = h_T^{(L)} W^{(k)} + b^{(k)}$ trained for offsets (k=1..K).
- **Self-speculative decoding**: Draft (K) tokens using multi-heads; verify with the base next-token head; accept the matching run.
- **Weight tying**: Reusing the input embedding matrix (transposed) as the output projection.

5. Gaps & Open Questions from the Literature

1. **Separation of representation vs. readout.** Probes can decode future info, but does the **LM's own head** exploit it?
2. **Causality vs. correlation.** High probe accuracy or neuron-target correlations don't prove that the base model *uses* these features during generation.
3. **Training recipes.** How much of multi-token benefit arises from the **auxiliary losses** vs. decoding **strategy** (self-speculation), and how does this scale with model size and domain?

6. Minimal Planned Experiments (Phase 1)

Objective. Test whether **independent auxiliary heads** attached to a **pretrained LM** can produce reliable (t+2) predictions **without updating** the trunk.

Plan. 1. **Model**: Choose a small pretrained decoder-only LM (e.g., GPT-2 small or a compact Pythia).
2. **Heads**: Attach **two auxiliary linear heads** $k \in \{1, 2\}$ on $h_T^{(L)}$ (one predicting (t+1) as a sanity baseline, one predicting (t+2)).
3. **Training scope**: **Freeze the entire trunk** (embeddings, attention, FFN, norms, base head). **Train only** the two new heads.
4. **Data**: Start with a manageable, tokenized domain (e.g., **The Pile** subsets, WikiText-103, or a code subset) to get quick signal.
5. **Loss**: Cross-entropy per head; track head-wise accuracy and **pass@k** for (k {1,5}) at offset (+2).
6. **Evaluation**: - **Probe-style**: Compare learned head performance to a separate linear probe trained on the same $h_T^{(L)}$ and target (t+2).
- **Ablations**: Different layers' $h_T^{(\ell)}$ as the readout; with/without weight tying; vary context length.

- **Sanity**: If (+2) head beats a naive "copy top-1 next-token twice" heuristic, that indicates real look-ahead signal in $h_T^{(L)}$.

Why this first. It isolates the **readout** question: *given the frozen representation learned for next-token prediction, how much (t+2) information is already extractable by a simple head?* Only if this is promising do we proceed to **full Meta-style** joint training and speculative decoding.

7. (Deferred) Phase 2: Toward Meta-style Training

If Phase 1 shows useful signal: - Unfreeze and train **K heads** jointly (accumulate gradients sequentially for memory).
- Implement **self-speculative decoding** and measure **speed-quality trade-offs** vs. vanilla decoding.
- Expand benchmarks (e.g., code tasks where Meta reported strongest gains).

8. Expected Contributions

- A clear, replication-oriented **review** of look-ahead diagnostics (probes, correlations) and multi-token mechanisms (heads, speculative decoding).
- **Empirical evidence** on how much multi-step information is linearly extractable from **frozen** pretrained representations.
- Practical guidance on when to invest in full multi-token training vs. lightweight readout heads.

9. Limitations & Risks

- **Frozen-trunk ceiling**: If $h_T^{(L)}$ wasn't shaped for (t+2), head-only training may underperform; negative results remain informative.
- **Dataset effects**: Domains like code may show stronger gains than free text; careful dataset selection matters.
- **Compute**: Even small-scale experiments need solid batching/tokenization; we'll keep models/datasets modest initially.

10. Conclusion

Recent work shows that LLM hidden states can contain future-relevant information and that multi-token prediction can improve efficiency and, in some domains, quality. Our first step is deliberately modest: attach two auxiliary heads to a frozen pretrained model and test whether (t+2) predictions are already linearly recoverable. Positive results would motivate a move toward full multi-token training and self-speculative decoding.

References

- Wu, X., et al. (2024). *Do Language Models Think Ahead?* arXiv:2404.00859.
- Meta AI (2024). *Scaling Multi-Token Prediction with LLMs.* arXiv:2404.19737.