# Exploring Myopia and Multi-Token Prediction in Transformer-Based Language Models

Harley T. Gribble

2025-09-25

## Abstract

Large Language Models (LLMs) are trained with next-token prediction objectives, yet recent work suggests they may internally represent information about **future** tokens beyond what is immediately required for prediction. Wu et al. (2024) argue that transformer models exhibit **non-myopic behavior**, encoding "look-ahead" information even when gradients are locally restricted. In parallel, Meta AI (2024) demonstrated that explicitly training models to predict **multiple future tokens in parallel** can improve inference efficiency and sometimes quality.

In this work, we reproduce and extend ideas from both perspectives by attaching **multi-token auxiliary prediction heads** to a pretrained GPT-2. In our first experiment, we froze the transformer trunk and trained auxiliary heads to predict tokens at offsets $t{+}1$, $t{+}2$, $t{+}3$ using a small slice of WikiText-2 (about **2.30M** real tokens for training and **0.24M** for validation; sequences padded to length 256). These **locally trained heads** partially recovered predictive ability (e.g., acc@1 = 0.305 for $t{+}1$) but lagged behind vanilla GPT-2 (acc@1 = 0.327), with larger gaps at $t{+}2$ and $t{+}3$. This mirrors the **myopic degradation** highlighted by Wu et al., where future-token prediction suffers without shared representation learning in the trunk.

In a second experiment, we **cloned the pretrained GPT-2 output head** to initialize the $t{+}1$ auxiliary head and **jointly trained** $t{+}1$ and $t{+}2$ heads (trunk still frozen). Performance at $t{+}1$ **exceeded baseline GPT-2** (acc@1 = 0.363 vs. 0.328), suggesting that **fine-tuning output layers alone can sharpen next-token decisions** even with a fixed trunk. However, accuracy at $t{+}2$ remained far below baseline autoregressive rollout (acc@1 = 0.156 vs. 0.331). A **McNemar exact test** on paired top-1 correctness indicated these differences are statistically significant (e.g., $p \approx 8 \times 10^{-93}$ for $t{+}1$; $p \ll 10^{-10}$ for $t{+}2$).

Together, our results support three takeaways:
(1) GPT-2 hidden states already contain signals useful for future tokens;
(2) auxiliary multi-offset heads can extract some of that signal; but

(3) **without backpropagating into the trunk**, representation sharing is limited, and future-token prediction underperforms vanilla autoregression. Ongoing work will explore **joint multi-head training** (Meta-style), **self-speculative decoding** for efficiency, and **probing analyses** (Wu-style) to better characterize how "look-ahead" information is organized inside transformer representations.

---

# 1. Introduction

Transformer-based language models are the foundation of modern NLP. By learning from massive corpora, they can generate fluent text and capture deep contextual patterns. However, fundamental questions remain about *how* they represent and utilize information across time steps, especially during autoregressive generation where each new token depends on all previous ones.

Two threads of recent research motivate this work:

1. **Myopia in language models (Wu et al., 2024)** — do models only learn to predict the *next* token, or do they implicitly capture information useful for future tokens beyond the immediate prediction task? Wu et al. show evidence that even without being explicitly trained to think ahead, language models still encode "breadcrumbs" of future tokens in their hidden states.
2. **Multi-token prediction (Meta AI, 2024)** — can we modify model architectures and training objectives to *explicitly* predict multiple future tokens in parallel, improving efficiency and generation stability? Meta demonstrated that adding auxiliary prediction heads allows models to generate multiple tokens per forward pass, enabling speculative decoding and faster inference.

Our work builds directly on these two ideas. **We explore whether auxiliary prediction heads can anticipate future tokens using only the existing internal representations of a pretrained transformer model.** Unlike Meta's approach, which retrains both the model trunk and the auxiliary heads jointly, we begin by freezing the base transformer (GPT-2) and training only additional prediction heads for future tokens. This setup resembles the *myopic constraint* studied by Wu et al., where no information from the future can update earlier layers of the model during training.

By doing so, we aim to answer the following questions:

- *How much future information is already present in a pretrained GPT-2 model's hidden states?*
- *Can auxiliary linear heads learn to extract future-token structure without modifying the model trunk?*

- *How does this approach compare to standard autoregressive prediction in practice?*
- *Do auxiliary heads produce meaningful predictions even when they are less accurate?*
- *What trade-offs exist between myopic learning constraints and explicit multi-token prediction?*

To investigate these questions, we ran experiments using GPT-2 with additional prediction heads trained to predict $t+1$ and $t+2$ tokens simultaneously. We evaluated accuracy, loss, and statistical significance against a vanilla GPT-2 baseline using McNemar's test. Early results show that while a frozen-trunk model cannot outperform full autoregressive generation at future offsets, it *can* learn partially meaningful multi-token structure—with surprising behavioral patterns that resemble both Wu et al.'s myopic findings and Meta's multi-token observations.

The remainder of this paper expands on the theoretical background, related work, experimental setup, and emerging results from our ongoing study.

---

## 2. Background

### 2.1 Neural Networks and Transformers

At their core, LLMs are neural networks. A simple feed-forward neural network computes:

$$y = f(Wx + b)$$

where $x$ is the input vector, $W$ are learned weights, $b$ is bias, and $f$ is a nonlinear activation (e.g., ReLU, GELU). Unlike simple networks, transformers operate on **sequences of tokens**. Each token is mapped to a learned embedding vector, and positional encodings are added so the model can reason about word order.

Transformers extend neural networks with **self-attention**, which allows each token to attend to other tokens in the sequence to gather contextual information. Hidden states are projected into **queries, keys, and values**:

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V$$

and combined using the attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

In **decoder-only models** like GPT-2, a **causal mask** is applied so positions can only attend to previous tokens, enforcing left-to-right prediction. Each transformer block also includes residual connections and Layer Normalization:

$$\text{Block}(H) = H + \text{MHA}(\text{LayerNorm}(H)) + \text{FFN}(\text{LayerNorm}(H))$$

After $L$ layers, the final hidden state $h_T^{(L)}$ is projected to vocabulary logits:

$$z = h_T^{(L)} W_{\text{out}} + b_{\text{out}}$$

which are converted to next-token probabilities:

$$P(\text{token} = v) = \frac{e^{z_v}}{\sum_j e^{z_j}}$$

---

## 2.2 Autoregressive Training with Cross-Entropy Loss

Transformer language models are trained using the **autoregressive next-token prediction objective**, also known as causal language modeling. Given a sequence of tokens $(x_1, x_2, \ldots, x_T)$, the model learns:

$$P(x_T \mid x_1, x_2, \ldots, x_{T-1})$$

Training uses **cross-entropy loss** at every position in the sequence:

$$\mathcal{L} = -\sum_{t=1}^{T-1} \log P(x_{t+1} \mid x_{\leq t})$$

Gradients from this loss are propagated backward through all model weights. This allows information about future tokens to influence earlier layers during training—an important detail when discussing **myopia** later in the paper.

---

### 2.3 Inference and Attention to Future Tokens

At inference time, tokens are generated one by one: each newly predicted token is appended to the context, and the process repeats. Interestingly, research suggests that hidden states sometimes encode information about upcoming tokens even before they are generated—this motivates the study of **multi-token prediction** and **myopia**, which we explore in later sections.

---

## 3. Related Work

### 3.1 Wu et al. (2024): Myopia in Transformers

Wu et al. investigated whether language models *implicitly* plan ahead or rely purely on next-token prediction. They defined **myopia** as a restriction that prevents a model from learning using information from *future* tokens during training. Their goal was to determine whether transformer representations naturally contain information about future text, even without being trained for it.

To study this, they compared:

- **Vanilla models** — trained normally with full backpropagation, where future tokens influence earlier hidden states through the loss.
- **Myopic models** — trained with **truncated gradient flow**, meaning only the most recent layer or token position had its gradients updated. Earlier layers were prevented from learning from future context.

They also introduced two types of **probes** on frozen model layers:

- **Linear probes** — trained on hidden states to predict $t + k$ tokens directly.
- **Neuron correlation analysis** — measured whether any *individual neuron* correlated with a future token label, as a sanity check that probes were not hallucinating structure.

They additionally compared against a **bigram baseline**, which predicts based on local statistics only. If bigram performance matched probe performance, it meant the model was behaving locally rather than encoding long-range structure.

**Findings**: - Even myopically trained models encoded future token information in their hidden states. - In small models, future information appeared only as a side effect of next-token prediction. - In larger models, **representations contained explicit future structure** — evidence of "implicit planning." - They defined a measurable **myopia gap** — the difference in performance between vanilla and myopic models.

This work demonstrated that transformer hidden states are not strictly myopic, even when trained to be so.

---

### 3.2 Meta AI (2024): Multi-Token Prediction

Meta challenged the standard autoregressive next-token paradigm by training models to predict **multiple future tokens in parallel**. Instead of a single output head predicting $t + 1$, they added $n$ additional linear heads to predict $t + 2, \ldots, t + n$:

$$\mathcal{L} = \sum_{k=1}^{n} \mathcal{L}_{t+k}$$

Unlike our approach of freezing the transformer trunk, Meta **updated all transformer weights** using the combined loss from all heads. To save memory, they did not compute gradients for all heads simultaneously—they used **sequential head training**, accumulating gradients before a backward pass.

They also introduced **self-speculative decoding**: generate $n$ tokens in one forward pass using the prediction heads, then verify them with a standard autoregressive step. This significantly improved inference speed.

**Findings**: - Multi-token prediction improved training efficiency and accelerated inference. - Performance gains were strongest for **code generation** and compositional tasks. - Larger models occasionally favored vanilla next-token loss, suggesting multi-token training changes internal representation learning. - They hypothesized that encouraging future awareness reduces "derailment" errors—bad decisions that compound over long sequences.

Meta's work provides a foundation for methods that make language models explicitly **look ahead**.

---

## 4. Methods

We conducted a series of controlled experiments to evaluate whether auxiliary future-token prediction heads can extract useful information from frozen GPT-2 representations. Our work evolved across two experimental designs.

---

### 4.1 Base Model

We use the pretrained GPT-2 small model (124M parameters) from Hugging Face. GPT-2 is a left-to-right causal transformer trained with next-token prediction. Given an input sequence of tokens, it outputs logits for the next token $t + 1$.

The transformer **trunk** (all self-attention and feedforward layers) was **frozen** in all of our experiments to isolate the contribution of future-token heads without altering internal representations.

---

### 4.2 Auxiliary Heads

We attach lightweight linear classifiers (heads) to GPT-2's final hidden layer to predict future tokens directly from hidden states:

- Head 1: predicts $t + 1$
- Head 2: predicts $t + 2$

In our **first experiment**, both heads were randomly initialized and trained from scratch.

In our **second experiment**, we **replaced Head 1** with a **copy of GPT-2's original next-token head** (its pretrained language modeling head). This allowed Head 1 to begin with strong next-token performance while being co-trained with the $t + 2$ head.

All transformer parameters remained frozen.

---

### 4.3 Training Objective

Both auxiliary heads shared a multitask cross-entropy loss:

$$\mathcal{L} = \mathcal{L}_{t+1} + \mathcal{L}_{t+2}$$

where each loss term is:

$$\mathcal{L}_{t+k} = -\sum_{t} \log p(x_{t+k} \mid h_t)$$

Only the auxiliary head weights received gradient updates.

---

### 4.4 Dataset

We trained on the WikiText-2 corpus (raw version) using a fixed-length context of 256 tokens. For compute efficiency, we used a **subset** of approximately **2.3 million real tokens** for training and **240 thousand tokens** for validation.

Tokenization used GPT-2's byte-pair encoding (BPE) tokenizer.

---

### 4.5 Evaluation Metrics

We compared the auxiliary-head model against frozen GPT-2 in two ways:

#### Quantitative Metrics

- **Hits@1**, **Hits@3**, **Hits@5** for $t+1$ and $t+2$
- **Cross-entropy loss**
- **Statistical significance** using **McNemar's exact test** over paired predictions

#### Qualitative Analysis

To understand model behavior beyond accuracy: - We logged **top-k token predictions** from both vanilla GPT-2 and aux heads - We compared predictions **inside real text contexts** - We analyzed **semantic plausibility** even when predictions were incorrect

---

### 4.6 Baseline vs. Auxiliary Comparison

| Condition | Description |
|---|---|
| **Baseline** | Vanilla GPT-2 predictions using teacher forcing and normal autoregression |
| **Auxiliary Heads** | GPT-2 trunk frozen, linear future-token heads trained |

Two configurations were tested: 1. **Independent heads** (random init for both $t+1$ and $t+2$) 2. **GPT-2 initialized head** (clone of pretrained $t+1$ head + trainable $t+2$ head)

---

# 5. Results

I conducted two main experiments using auxiliary future-token prediction heads attached to a frozen GPT-2 model. All experiments used the WikiText-2 raw dataset with approximately 2.3 million real training tokens after padding removal and 240k validation tokens.

---

### 5.1 Baseline (vanilla GPT-2 performance)

Before training any auxiliary heads, I evaluated the pretrained GPT-2 model using teacher-forced logits to measure its ability to predict future tokens:

| Offset | Hits@1 | Hits@3 | Hits@5 | Loss | Tokens (N) |
|---|---|---|---|---|---|
| $k = +1$ | 0.328 | 0.478 | 0.543 | 3.893 | 52,902 |
| $k = +2$ | 0.331 | 0.483 | 0.549 | 3.847 | 52,258 |

This establishes the baseline performance for comparison.

---

### 5.2 Experiment 1 – Independent auxiliary heads (random initialization)

In the first experiment, I trained two auxiliary linear output heads (for $t+1$ and $t+2$ prediction) attached to a **frozen GPT-2 trunk**. These heads were randomly initialized and trained independently of the original GPT-2 output head.

| Offset | Hits@1 | Loss |
|---|---|---|
| $k = +1$ | 0.305 | 5.702 |
| $k = +2$ | 0.155 | 7.240 |

**Observation:**
The $t+1$ head recovered approximately **93%** of baseline next-token performance, even though the trunk was frozen. However, performance on $t+2$ dropped significantly, indicating that the GPT-2 final hidden state contains only **limited usable information about future tokens** without autoregressive conditioning or trunk updates.

---

### 5.3 Experiment 2 – Cloned $t{+}1$ head + future $t{+}2$ head

To better isolate future-token prediction, I modified the approach by cloning the pretrained GPT-2 LM head and using it as the $t+1$ auxiliary head. This ensured that the $t+1$ prediction began with the pretrained next-token capabilities already present in GPT-2. I then co-trained this cloned head together with the $t+2$ auxiliary head.

| Offset | Hits@1 | Loss |
|---|---|---|
| $k = +1$ | **0.363** | 3.970 |
| $k = +2$ | 0.156 | 7.252 |

**Observation:**
Unlike Experiment 1, the cloned $t+1$ head actually **outperformed the vanilla GPT-2 head** (0.363 vs 0.328 Hits@1). Even without training the transformer trunk, the cloned head fine-tuned to the dataset and improved next-token accuracy. However, performance at $t+2$ remained far below the autoregressive baseline.

---

### 5.4 Statistical comparison (McNemar test)

To assess whether differences in prediction accuracy between auxiliary heads and the baseline GPT-2 were statistically significant, I used the **paired McNemar exact test** on per-token correctness:

| Offset | $p$-value | Conclusion |
|---|---|---|
| $k = +1$ | $7.98 \times 10^{-93}$ | Highly significant |
| $k = +2$ | 0.0 | Highly significant |

Small differences in accuracy are meaningful over tens of thousands of token comparisons.

---

### 5.5 Qualitative prediction analysis

To better understand prediction behavior, I inspected top-$k$ predictions from both models. I found that while GPT-2 maintained **context-sensitive predictions**, the auxiliary future head often produced **high-frequency but semantically shallow tokens** such as `"the"`, `"of"`, `"and"`, `"in"`.

Example ($k = +2$):

```
Context: "... with a body length up to 60 centimetres (24 in) and"
Gold target: "weigh"
Vanilla GPT-2: "weigh"
Aux head: "The" (Top-5 included: "are", "is", "the", "it")
```

This suggests that **future-token prediction ($k > 1$) is harder without autoregressive rollout**, and the auxiliary head falls back to **syntactic completions** rather than semantic ones.

---

### 5.6 Summary of findings

- GPT-2 already encodes weak future-token information in hidden states.
- The $t + 1$ auxiliary head can match or even **exceed baseline GPT-2 performance** when initialized properly.
- Future prediction at $t+2$ from a single hidden state is **much weaker** than autoregressive GPT-2 prediction.
- Auxiliary heads trained in isolation tend toward **generic filler tokens**, not meaningful continuation.
- These findings partially support Wu et al. (2024) regarding **implicit forward signal**, but also show the **limits of static hidden states** without sequence rollout.

---

## 6. Discussion

The results from these preliminary experiments offer insight into how transformer-based language models represent future-token information and how prediction heads interact with frozen vs. trainable model components.

### Relation to Wu et al. (2024)

My setup using **auxiliary heads on a frozen transformer trunk** mirrors the **myopic model** condition described by Wu et al. The model is prevented from updating deeper contextual representations, and only the output layers are trained. Consistent with their findings:

- Even without gradient flow through the trunk, the model retained **non-zero ability to predict future tokens** ($k > 1$).
- This suggests that GPT-2 hidden states **implicitly encode information relevant to upcoming tokens**, even though it was never explicitly trained to think ahead.
- However, similar to Wu et al.'s reported **myopia gap**, my auxiliary $t + 2$ predictions performed far worse than the baseline autoregressive GPT-2, supporting the idea that **hidden states are locally predictive but not future-aware enough on their own**.

### Relation to Meta (2024)

My work also connects to Meta's multi-token prediction method but differs in a crucial way:

| Aspect | Meta Approach | My Experiment |
|---|---|---|
| Trunk updated? | Yes | No (frozen) |
| Multi-head training | Yes | Yes |
| Self-speculative decoding | Yes | No |
| Objective | Future prediction + speed | Analyze future info in hidden states |

Meta found that jointly training the output heads **with trunk updates** improved multi-token prediction quality. In contrast, my $t + 2$ head struggled because the frozen trunk was never optimized to support multiple future outputs. This suggests:

> **Future prediction is not just an output-layer problem — it requires trunk adaptation.**

### Effects of Cloning the GPT-2 Head

Cloning the pretrained GPT-2 head as the $t+1$ head (Experiment 2) **significantly improved behavior** compared to random initialization:

- The cloned $t + 1$ head **outperformed GPT-2's original next-token prediction** (+3.5% Hits@1).
- This suggests the auxiliary training may act as a mild **domain adapter**, improving performance on WikiText-2.
- However, this improvement did **not transfer to $t + 2$ predictions**, which still lagged behind the autoregressive baseline by more than 50%.

### Future Tokens Are Hard Without Rollout

The $t + 2$ predictions were often **grammatical but semantically weak**, favoring common words like "the," "of," or "and." This indicates:

- Without autoregressive context updates, the model falls back on **generic English continuation priors**.
- The transformer's hidden state **does not contain strong enough information beyond a single decoding step**.
- This aligns with the **information decay hypothesis** — hidden states are optimized for local prediction, not for long-range future supervision.

### Statistical Confidence

Through McNemar's test, I confirmed that even small differences in Hits@1 were **statistically significant** due to large sample sizes. This validates that the auxiliary heads are consistently weaker than GPT-2 at $k = +2$, and any apparent improvements are not due to noise.

---

**Conclusion of Discussion:**

These first experiments support the idea that GPT-2 is **partially myopic** — it shows some awareness of future tokens, but only enough for weak signal extraction. Predicting two or more steps ahead is fundamentally limited without **autoregressive rollout or joint training** like in Meta's paper. My findings reinforce that **future-aware modeling must modify the trunk**, not just stack output heads.

---

# 7. Future Work

The results to date demonstrate that frozen-trunk auxiliary heads can extract limited future-token information from GPT-2 hidden states, but performance drops sharply beyond $k = +1$. Building on this, I plan the following next steps:

## 7.1 Training Variants

- **Compare frozen vs. unfrozen trunk:** Allow gradient updates to flow partially or fully through the transformer trunk to test whether trunk adaptation improves $t + 2$ prediction quality, as suggested by Meta (2024).
- **Multi-token + next-token co-training:** Train both the original GPT-2 head and the $k = +2$ head jointly to explore whether the next-token objective stabilizes future-token supervision.

## 7.2 Myopia-Probe Connection

- **Linear probing analysis:** Apply linear probes to hidden states at various layers to measure how much $t + 2$ information is present before and after auxiliary training. This will allow direct comparison with Wu et al. (2024).
- **Information localization:** Determine which transformer layers encode future-token signal most strongly.

## 7.3 Prediction Analysis

- **Top-$k$ error inspection:** Expand qualitative evaluation by categorizing future-token errors (semantic vs. syntactic vs. generic fallback).
- **Prediction drift measurement:** Quantify how far $t + 2$ predictions deviate from the autoregressive baseline in embedding space.

## 7.4 Speculative Decoding

- **Implement self-speculative decoding:** Use the $k = +2$ head to draft tokens and verify them using the base GPT-2 head. This will test whether accuracy can be retained while reducing decoding steps.
- **Speed vs. accuracy tradeoff:** Compare decoding latency and perplexity against standard autoregressive GPT-2.

### 7.5 Scaling and Data

- **Larger models:** Replicate experiments on GPT-2 Medium and GPT-2 Large to study scaling effects.
- **Domain specialization:** Test whether future-token heads perform better on structured domains like **code** (e.g., The Stack dataset) or **math**.
- **Curriculum training:** Increase auxiliary supervision strength gradually (start with $k = +1$, then $k = +2$).

---

The overarching goal is to explore whether multi-token prediction can improve transformer planning and long-term coherence without sacrificing accuracy — and whether these gains require trunk-level representation learning or can emerge locally through auxiliary supervision.

---

## 8. Conclusion

This work explored the relationship between local prediction objectives (myopia) and explicit future-token supervision (multi-token prediction) in transformer-based language models. Prior work by Wu et al. (2024) demonstrated that even when trained myopically, language models retain some information about future tokens in their hidden states. Meta AI (2024) extended this idea by showing that explicitly training models to predict multiple tokens ahead can improve efficiency and stability during inference.

Building on these ideas, I implemented auxiliary future-token prediction heads on top of a pretrained GPT-2 model. Initial experiments froze the transformer trunk and trained linear auxiliary heads to predict tokens at offsets of t+1 and t+2 directly from the final hidden state. These auxiliary heads were able to recover a meaningful portion of next-token predictive accuracy without modifying the base GPT-2 weights, confirming that some future-token signal is inherently present in the hidden representations.

However, performance on predicting t+2 tokens remained substantially lower than the autoregressive baseline, even when initializing the t+1 head using GPT-2's pretrained output layer. Statistical analysis using McNemar's test confirmed that this performance gap was highly significant. Qualitative analysis of predictions revealed that auxiliary heads frequently produced syntactically valid but semantically generic or context-insensitive outputs, suggesting that frozen hidden representations are insufficient for deeper future-token reasoning.

These findings support two emerging hypotheses: (1) transformer hidden states do contain partial future-token information, consistent with Wu et al. (2024), but (2) effective future-token prediction likely requires adapting the underlying representations through joint training, as done in Meta's multi-token approach. The results also indicate that auxiliary heads may provide a low-cost method to exploit early planning behavior in small models, but their full potential may only be realized when coupled with shared gradient updates through the transformer trunk.

Future work will focus on joint training strategies, representation probing, scaling to larger models, and applying auxiliary supervision to structured domains such as code. The ultimate goal is to better understand how transformer models internally plan ahead and whether explicit multi-token supervision can enhance coherence, stability, and reasoning without sacrificing accuracy.

---

## References


- Wu, Y., et al. (2024). *Do Language Models Think Ahead?*
- Meta AI (2024). *Multi-Token Prediction Improves LLM Training and Inference Efficiency.*
- Vaswani, A., et al. (2017). *Attention Is All You Need.*