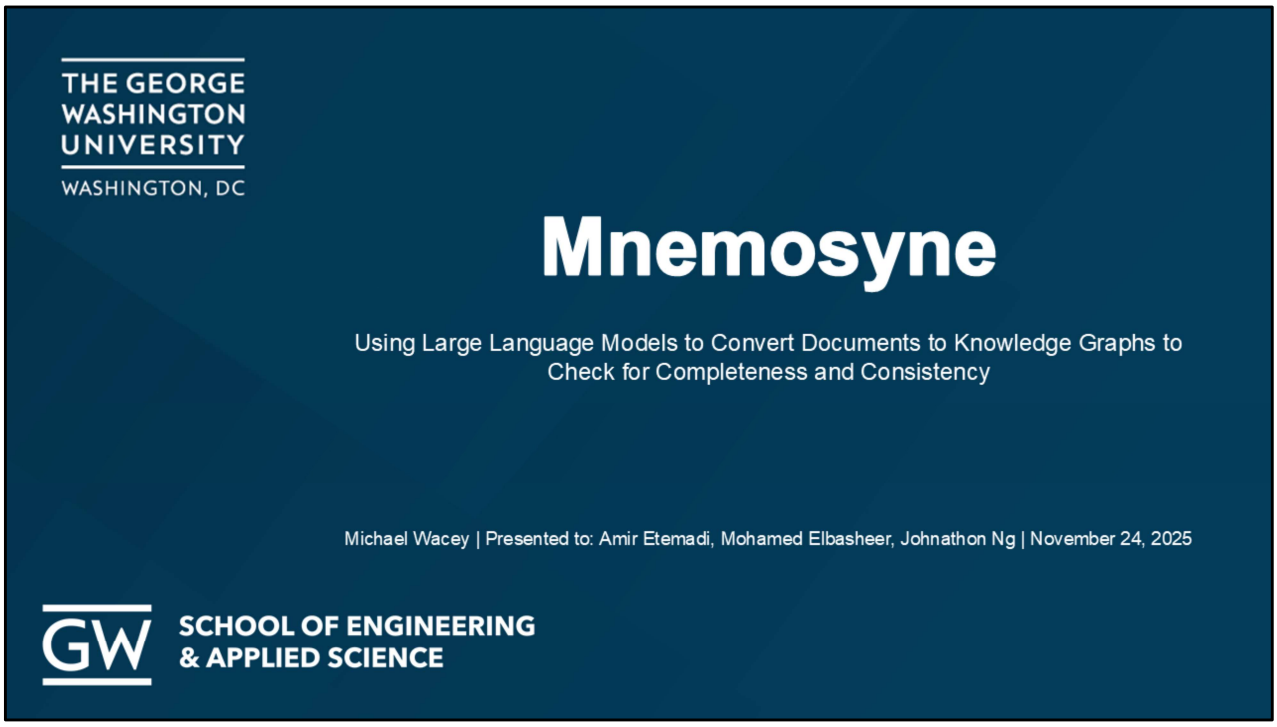
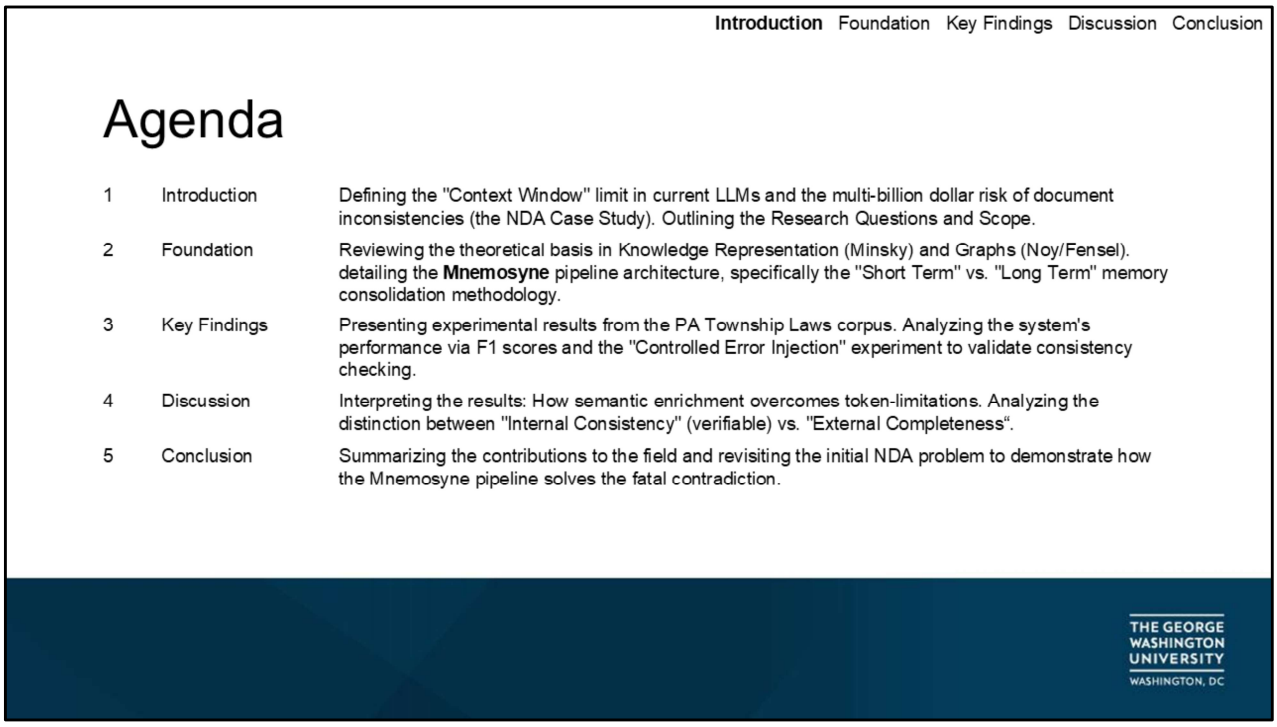


Transition: "Good afternoon. I am Michael Wacey."

Notes:

"Welcome to my presentation on Mnemosyne. In Greek mythology, Mnemosyne is the goddess of memory and the mother of the Muses. I chose this name for my research because, at its core, this praxis is about solving the problem of 'memory'—specifically, the limited context window—in Large Language Models. I am honored to present this work to my committee: Dr. Amir Etemadi, Dr. Mohamed Elbasheer, and Dr. Johnathon Ng."

Conclusion: "Today, I will demonstrate how we can turn static documents into active, queryable knowledge graphs to solve multi-billion dollar inconsistencies."



Transition: "My presentation today mirrors the structure of my written praxis, divided into five key sections."

Notes:

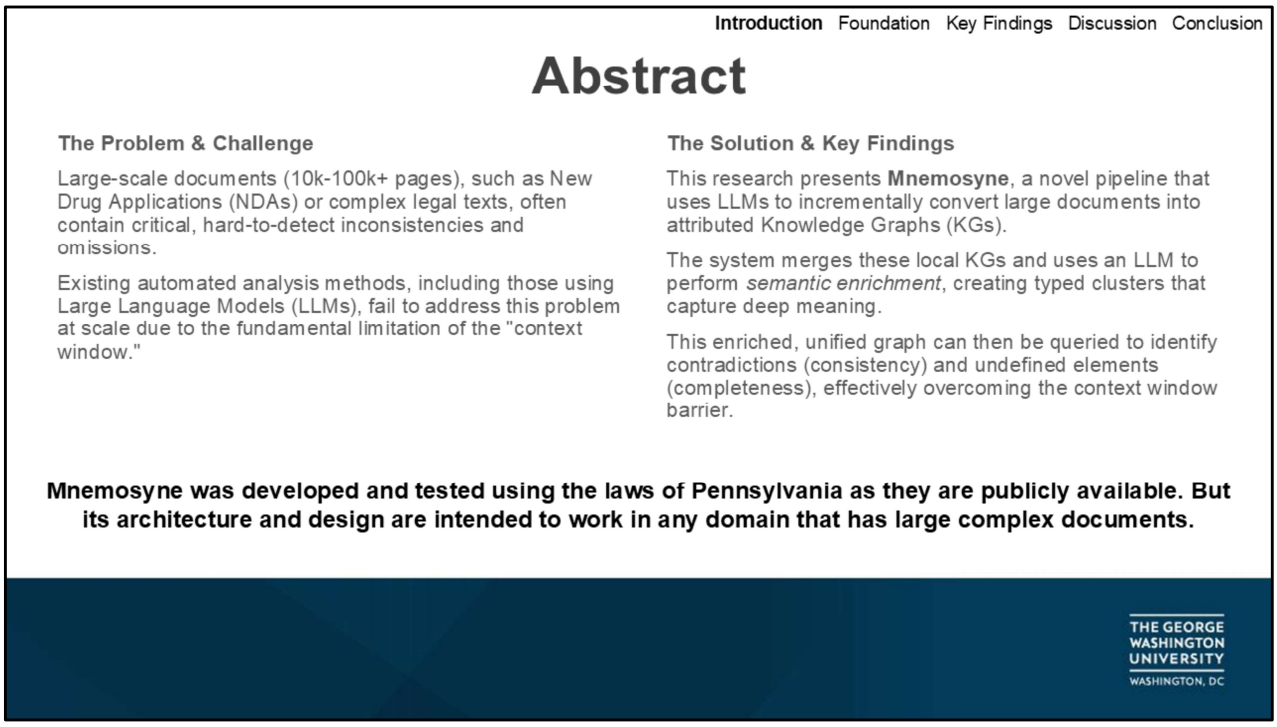
"**Introduction:** Here, I will outline the research questions and the scope of the study, focusing on the limitations of current LLMs."

"**Foundation:** I will explain the 'Short Term' versus 'Long Term' memory consolidation methodology used in the Mnemosyne system."

"**Key Findings:** I will present the data gathered from the PA Township Laws corpus and analyze the system's performance via F1 scores."

"**Discussion:** We will distinguish between 'Internal Consistency,' which is verifiable, and 'External Completeness'."

Conclusion: "And finally, in the **Conclusion**, I will close by summarizing the contributions to the field."



Transition: "This slide outlines the core conflict my research addresses."

Notes:

"The **Problem** is one of scale. In domains like pharmaceuticals or law, we deal with massive documents—NDAs often exceed 100,000 pages."

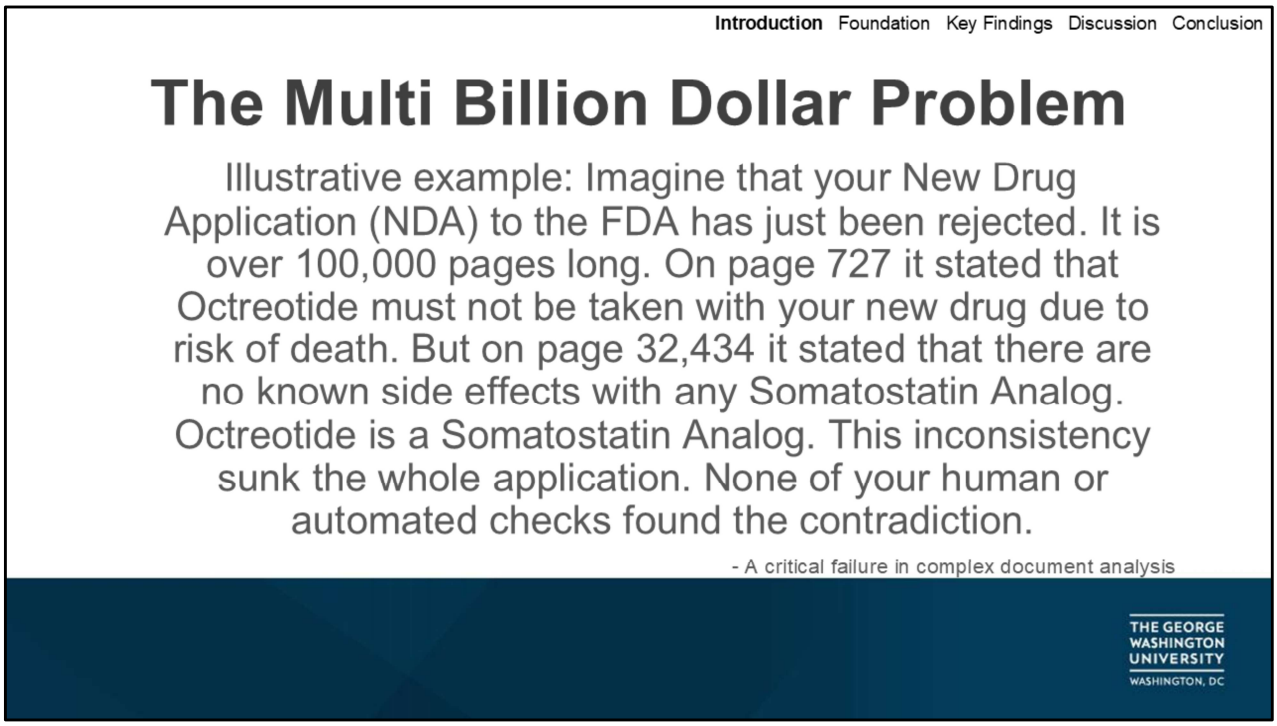
"Standard Large Language Models fail here because of the 'Context Window' limit."

"Essentially, the model forgets the beginning of the document before it reaches the end, making holistic analysis impossible."

"My **Solution** is a pipeline called Mnemosyne. Instead of forcing raw text into a limited window, Mnemosyne incrementally converts the document into an Attributed Knowledge Graph."

"By merging these graphs and using an LLM to perform semantic enrichment, we create typed clusters that capture the deep meaning of the text."

Conclusion: "This effectively solves the context problem, allowing us to query the graph directly to identify contradictions and missing information without needing the entire text in memory."



Transition: "To understand the stakes, I'd like to walk you through an illustrative nightmare scenario."

Notes:

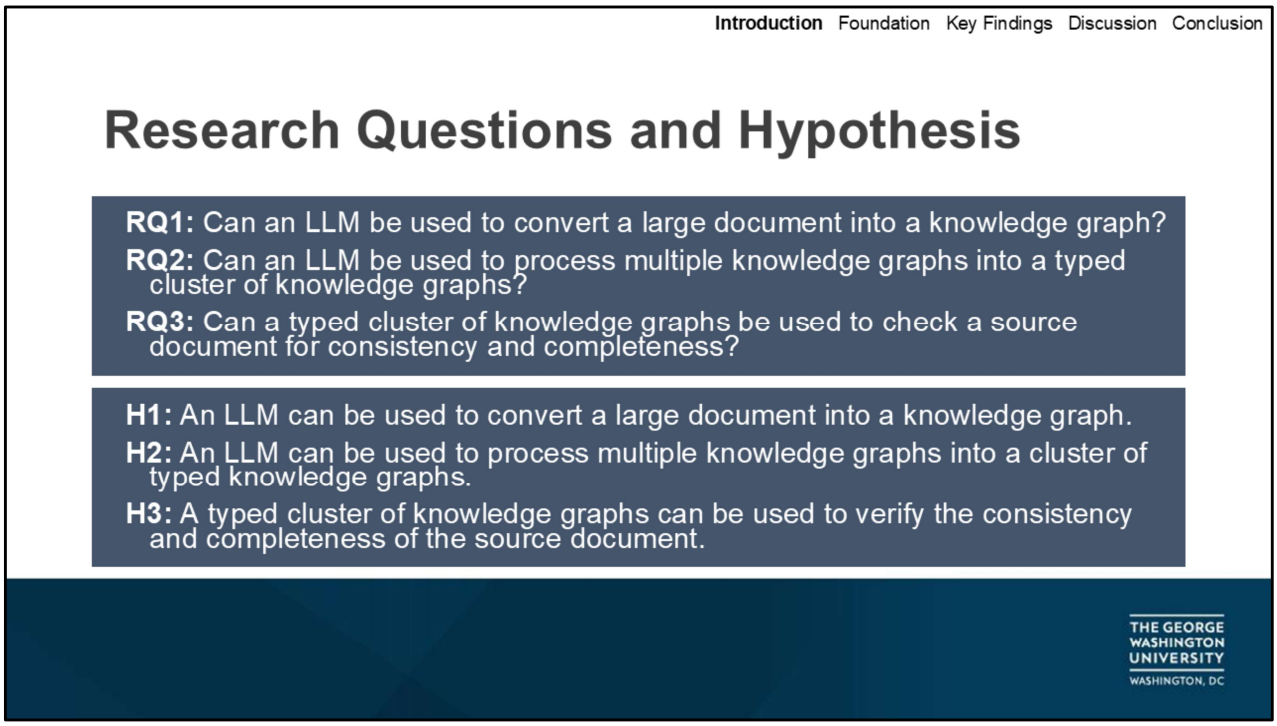
"Imagine you have just submitted a 100,000-page New Drug Application to the FDA. On page 727, your documentation correctly notes that **Octreotide** carries a risk of death if taken with your drug."

"However, 30,000 pages later, on page 32,434, a different section states there are 'no known side effects' with any **Somatostatin Analog**."

"The problem is that Octreotide *is* a Somatostatin Analog. This logical contradiction sinks the application."

"I want to be clear for the committee: This specific narrative is a synthesized example designed to isolate the logical error. In the appendix of my deck, I detail the messy, real-world rejection cases—such as Vanda Pharmaceuticals and PTC Therapeutics—that inspired this composite."

Conclusion: "I use this simplified example here to demonstrate the exact type of 'needle-in-a-haystack' inference failure that Mnemosyne is designed to detect."



Transition: "To guide this research, I established three specific questions."

Notes:

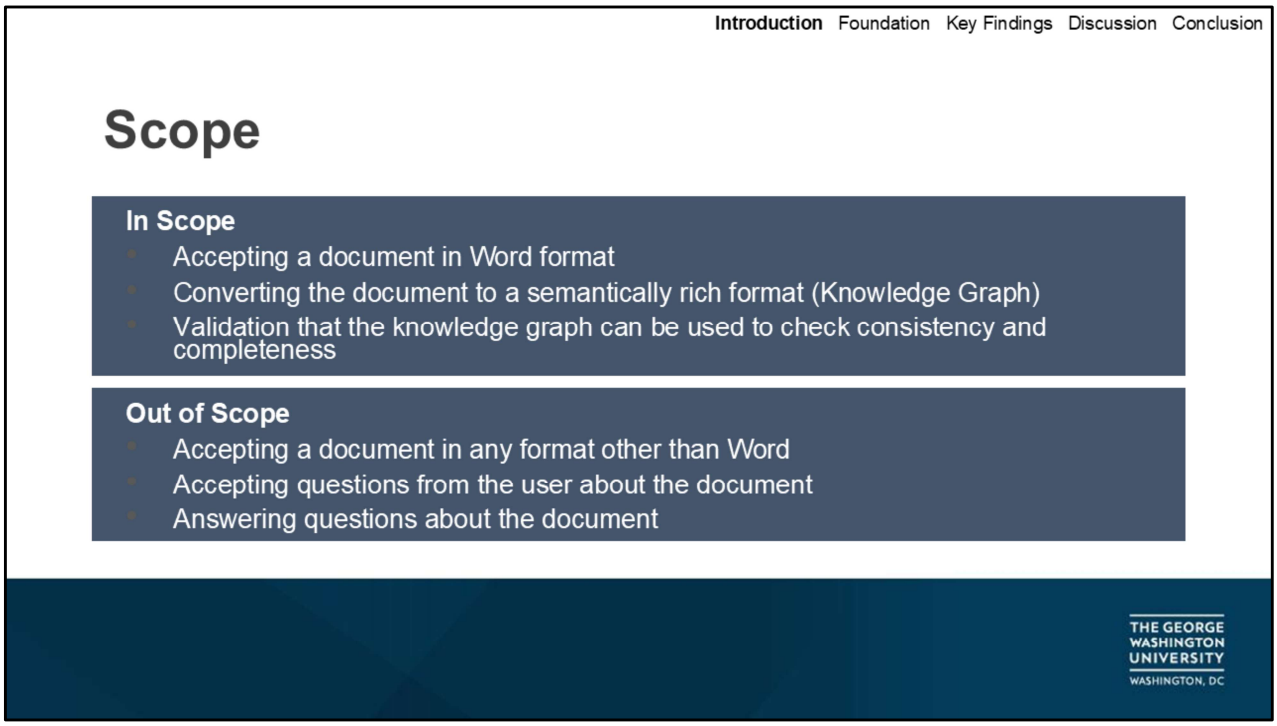
"**RQ1:** Can an LLM successfully convert a large, unstructured document into a structured knowledge graph?"

"**RQ2:** Can we then process multiple local graphs into a single, semantically enriched 'typed cluster'?"

"**RQ3:** And finally, can this enriched cluster be used to verify the consistency and completeness of the source document?"

"For my hypotheses, I posited the affirmative for each: that the pipeline would not only build these graphs but that the resulting structure would allow for mathematical verification of the text."

Conclusion: "I used my Township Law dataset as the proving ground to test these hypotheses."



Transition: "To ensure this research remained focused on the novel contributions regarding graph consistency, I established strict boundaries in consultation with Dr. Elbasheer and the Department Leaders."

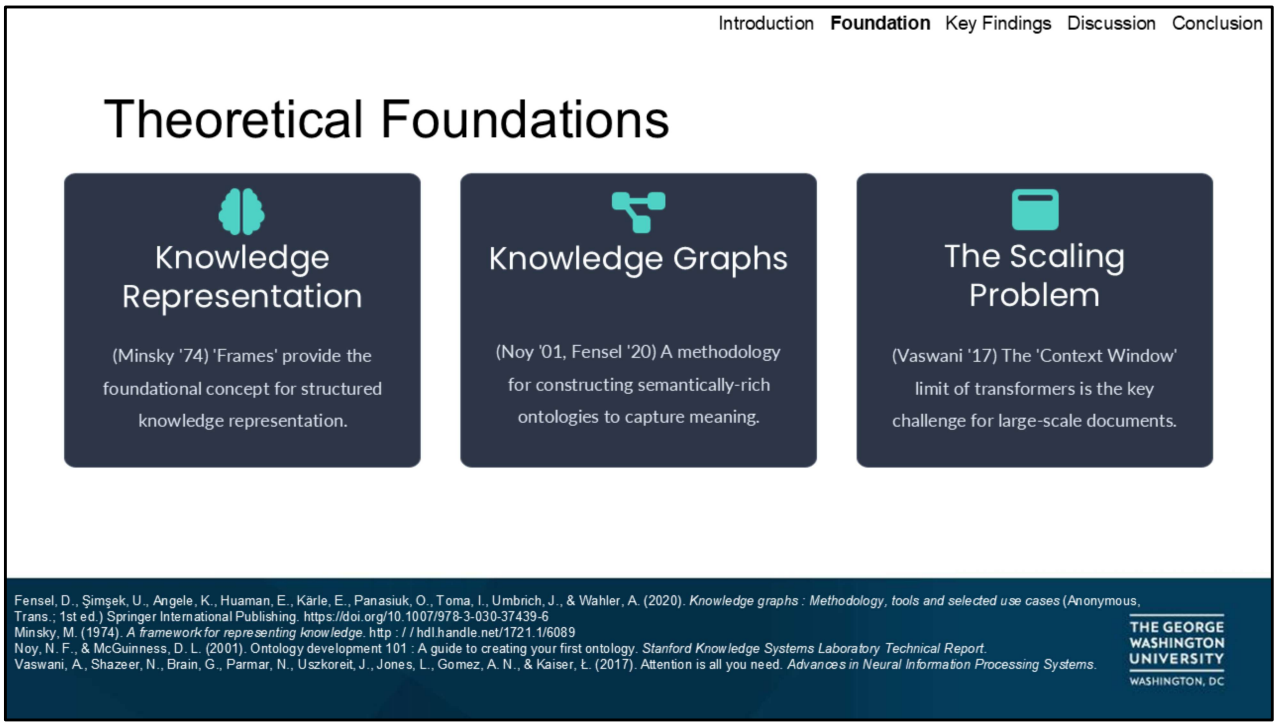
Notes:

"**In Scope:** We decided to focus strictly on the pipeline logic: ingesting clean text (specifically Word documents), converting it to a Graph, and validating the consistency checks."

"**Out of Scope:** Consequently, we explicitly excluded 'product' features like PDF parsing or a user-facing Chatbot interface."

"This scoping allowed me to devote my engineering efforts entirely to the semantic enrichment algorithms rather than solving solved problems like file parsing."

Conclusion: "This ensured the research targeted the unsolved problem of semantic consistency rather than the solved problem of file ingestion."



Transition: "My research rests on three theoretical pillars that define both the structure and the constraint of the system."

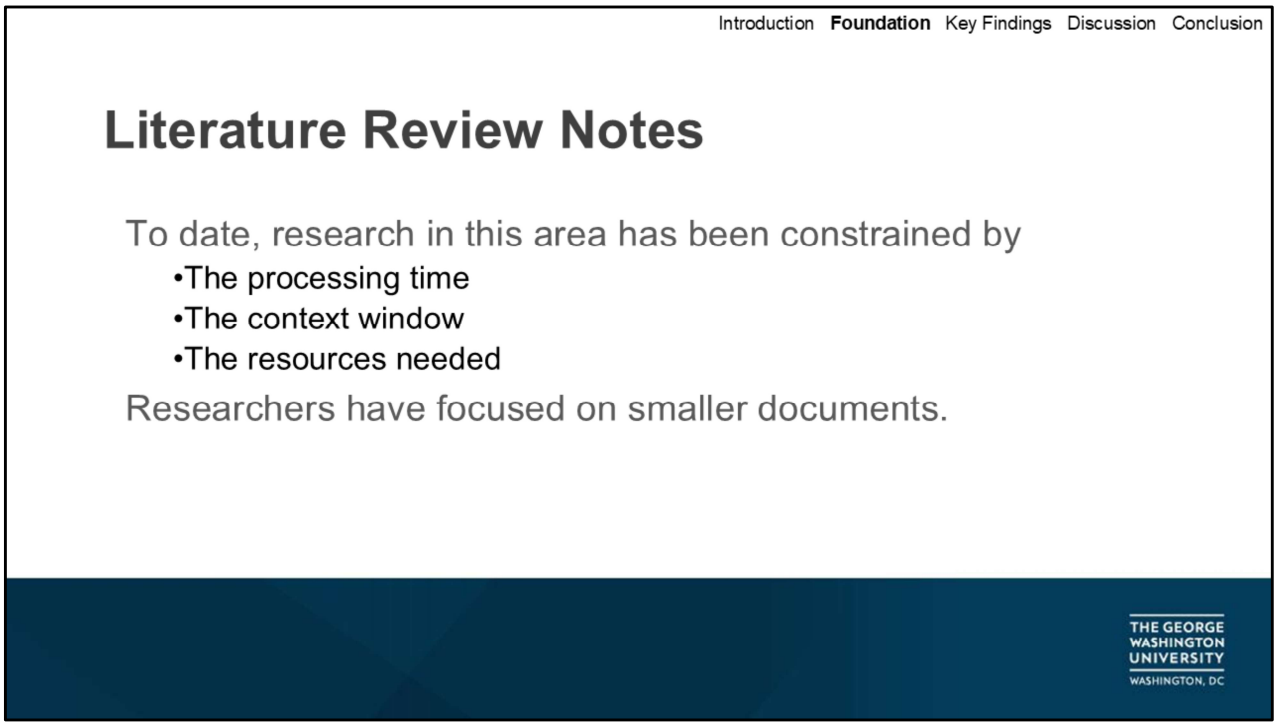
Notes:

"First is **Minsky (1974)**: His concept of 'Frames' is the basis for my system's 'Nodes.' He argued that knowledge isn't a stream of words, but distinct, structured units.“

"Second are **Noy & Fensel**: They provided the 'Knowledge Graph' methodology. If Minsky gave us the 'dots,' they gave us the 'lines'—the semantic relationships that turn isolated facts into understanding.“

"Third is **Vaswani (2017)**: He defined the 'Context Window.' This is our scaling constraint. Crucially, while windows have grown larger, we now face 'Attention Degradation,' where the model cannot maintain coherence over vast distances."

Conclusion: "Mnemosyne uses the structure of Minsky and Noy to provide the focus necessary to solve the limitations of Vaswani's context window."



Transition: "Having established the theoretical pillars, I conducted a review of current applications of LLMs in knowledge extraction. I found that despite the hype, the field is currently hitting a ceiling."

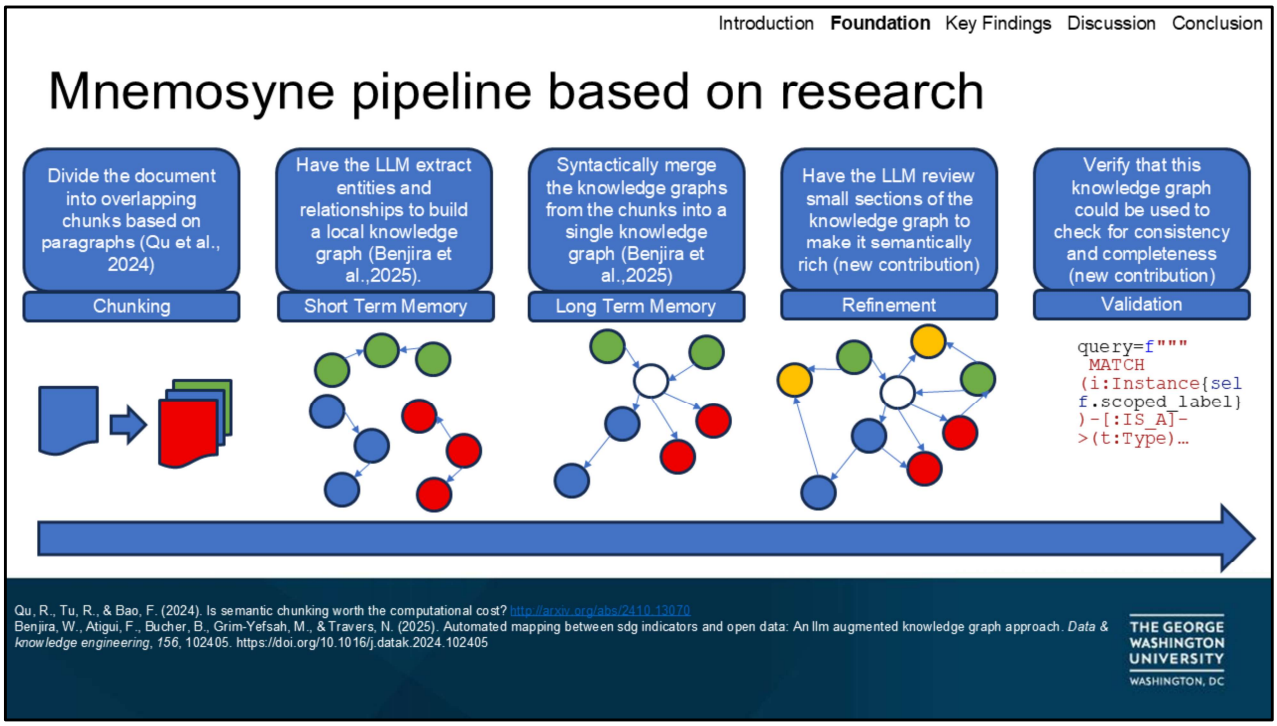
Notes:

"To date, practical research in this area has been severely constrained by three interconnected factors: **Processing Time**, the **Context Window** (specifically attention degradation), and **Resource Intensity**."

"Because of these constraints, the vast majority of researchers have focused on smaller documents—summarizing news articles or extracting facts from single abstracts."

"Very few have attempted to tackle the '100,000-page problem' in a way that preserves rigorous logic."

Conclusion: "This gap defined my design requirements. To handle an NDA, I needed a system that did not rely on a massive single context window and could manage resources efficiently through incremental processing."



Transition: "This slide illustrates how I translated the literature review into the Mnemosyne pipeline."

Notes:

“I want to clarify a detail regarding the Refinement stage (P5). In the written Praxis, I note that the *asynchronous* processing engine was disabled during experiments. To ensure precise timing measurements and deterministic output for this defense, I ran the Refinement logic **synchronously**—in batch mode—rather than as a background thread. This allowed me to capture the semantic enrichment metrics you will see on Slide 16, while maintaining strict control over the experimental variables.”

"I built the foundation of the system on recent methodologies: adopting **Qu et al.** for paragraph-based chunking and **Benjira et al.** for extracting entities into local Knowledge Graphs.“

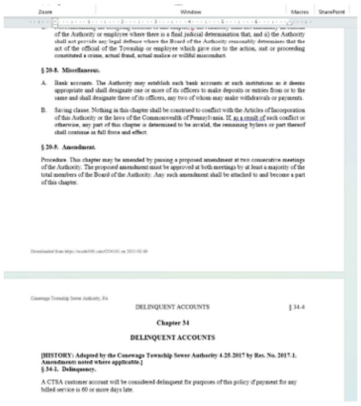
"However, existing methods stop there, resulting in a 'flat' graph. My novel contributions are the **Refinement** and **Validation** steps.“

“I utilize a hybrid approach: using Procedural Code for deterministic tasks like chunking, and LLMs only for probabilistic tasks like extraction.“

"The pipeline moves from **Short Term Memory** (a temporary JSON buffer) to **Long Term Memory** (the Neo4j database), and finally to **Refinement**, where the system revisits the graph to semantically enrich it."

Conclusion: "This Refinement step effectively turns a static map of words into a queryable structure that can actually check for consistency."

Chunking



Strategy	Context Preservation	Implementation Difficulty	Verdict
Fixed Number of Characters	• No context • Breaks could be meaningless	• Easy • Size can match the context window	• Too much loss of context
Fixed Number of Sentences	• Can lose context • Breaks could be meaningless	• Small units • Easy • Size needs to be adjusted to fit in Context window	• Context can still be lost
Fixed Number of Paragraphs	• Good Context Boundaries • Larger Units	• Easy • Size needs to be adjusted to fit in Context windows	• Good balance of context • Selected Approach
Meaningful Groups	• Only Context Boundaries	• Requires Manual Effort or LLM • Size needs to be adjusted to fit in Context windows → may require splitting	• Too much of a performance penalty

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Transition: "The first step in the pipeline is 'Chunking,' where we determined how to feed the document into the model."

Notes:

"During the early prototyping phase, I ran several heuristic tests using fixed character and sentence splitting.“

"I observed immediately that these methods were destructive—often cutting critical definitions in half or separating subjects from their verbs.“

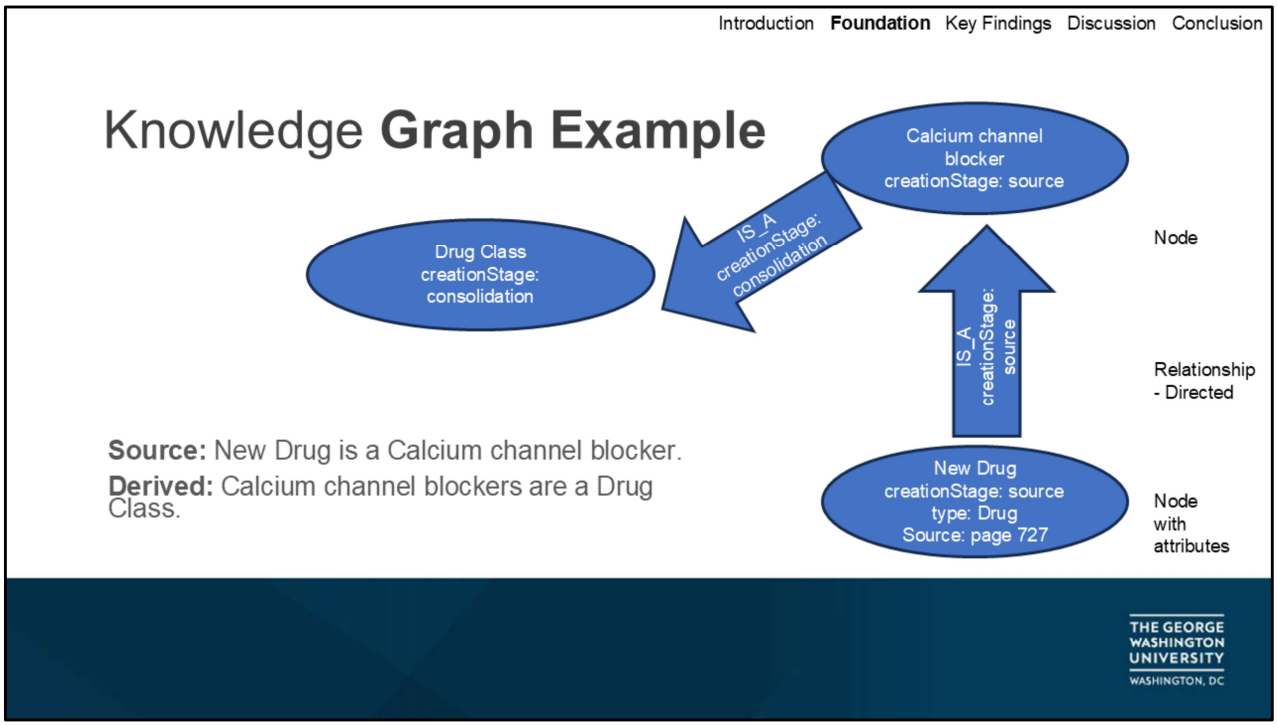
"While they were computationally cheap ('Easy'), the loss of semantic context was unacceptable for the downstream consistency checks.“

"Based on those early tests, I formalized the decision to use a **Fixed Number of Paragraphs**.“

"Paragraphs are natural, author-defined context boundaries."

Conclusion: "This approach provided the necessary semantic integrity without the massive performance penalty of using an LLM to pre-read the text.“

Research, literature review, critical for fitting context window.



Transition: "This slide illustrates the logic Mnemosyne uses to construct the graph, applied here to the Octreotide example."

Notes:

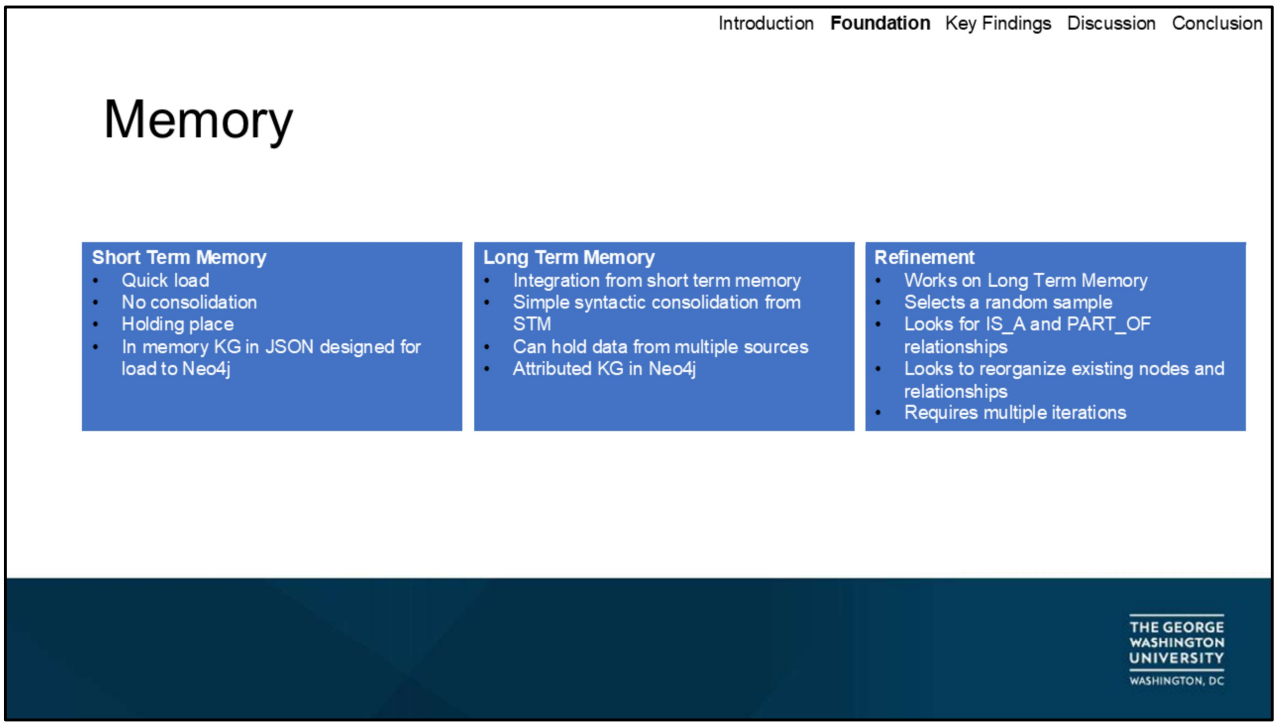
"At a basic level, the knowledge graph consists of **Nodes** and **Directed Relationships**."

"However, one specific innovation is in how the system assigns attributes. In this scenario, because the text explicitly mentions 'New Drug' and 'Calcium channel blocker', the system would mark those nodes as creationStage **Source**."

"But to check for consistency, the system needs to understand what 'Calcium channel blocker' *is*. During the consolidation phase, the logic dictates creating a new node called 'Drug Class' to link them."

"This new node is marked as creationStage **Consolidation**. Early on, I called this stage Consolidation. I latter realized it was really Refinement. But I never changed the code."

Conclusion: "This distinction is vital: it allows a human to later audit the graph and distinguish between facts derived explicitly from the source text and scaffolding generated by the AI. This can be used to check for hallucinations."



Transition: "To handle 100,000-page documents, I designed Mnemosyne with a two-stage memory architecture that mirrors human cognition."

Notes:

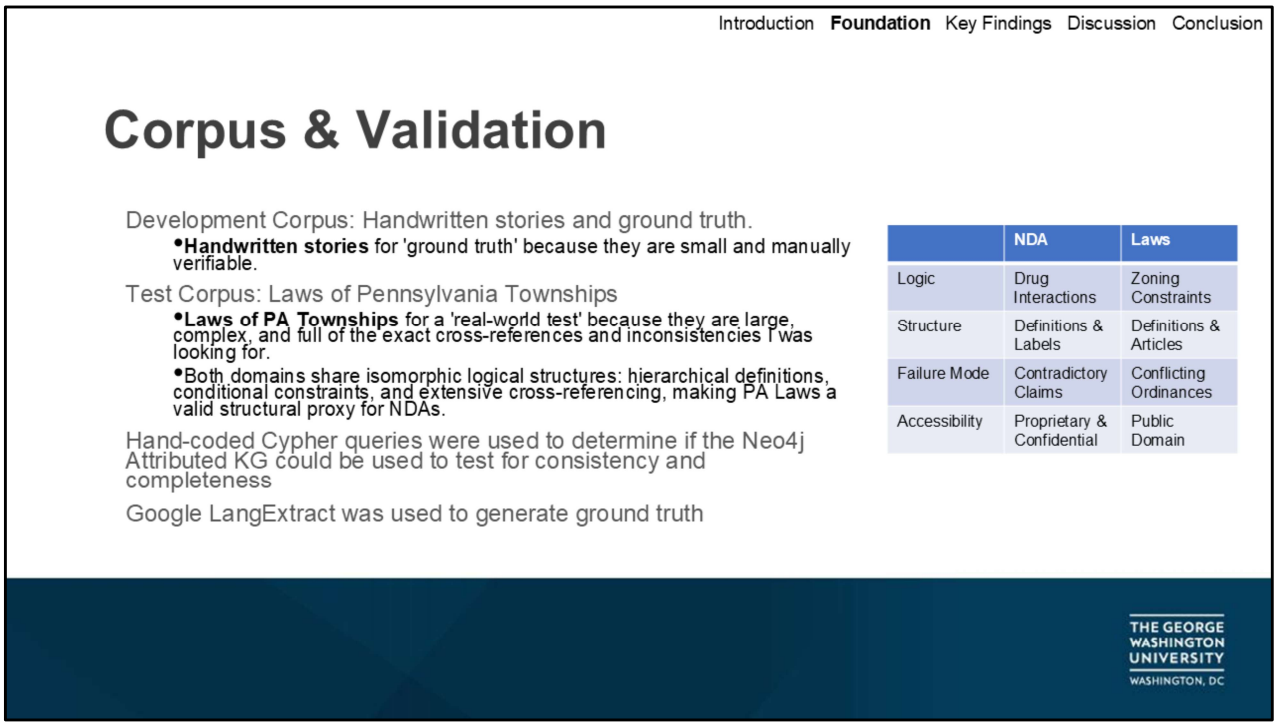
"First is **Short Term Memory**: This is the 'quick load' buffer. As the LLM reads each paragraph chunk, it extracts entities into a lightweight in-memory JSON format. There is no deep thinking here; it is just a holding place."

"Next is **Long Term Memory**: This is where we consolidate those chunks into a persistent Neo4j database. This allows us to hold data from multiple sources in a single repository."

"Finally, the **Refinement Phase**: This is the most critical step. The system revisits the Long Term Memory—much like a human consolidating memories during sleep."

"It selects random samples and specifically looks for IS_A and PART_OF relationships to reorganize and cluster the nodes."

Conclusion: "This process turns a flat list of extracted facts into a connected web of knowledge."



Transition: "To validate the pipeline, I utilized two distinct corpora."

Notes:

"First, for Development: I used **Handwritten Stories** with hand derived 'ground truth'."

"Second, for Real-World Testing: I selected the **Laws of Pennsylvania Townships**."

"I chose this legal dataset for two specific reasons:"

"**Availability:** NDAs are proprietary and highly confidential, making them impossible to secure for open research. Township laws are public domain."

"**Isomorphism:** While the vocabulary differs, the logic is identical. Both an NDA and a Township Law consist of definitions that constrain entities, and rules that are conditional on those definitions."

Conclusion: "If Mnemosyne can find a contradiction in a zoning law, it demonstrates the exact reasoning chain required to find a drug interaction contradiction in an NDA."

Transition: "With the pipeline built and the corpora selected, we can now look at the empirical evidence of how Mnemosyne performed."

Notes:

"Over the next few slides, I will present the experimental results gathered from the Pennsylvania Township Laws corpus. I have divided these findings into two categories."

"First, **Quantitative Analysis**: We will look at the F1 scores to evaluate how well Mnemosyne extracted the knowledge graph compared to the ground truth."

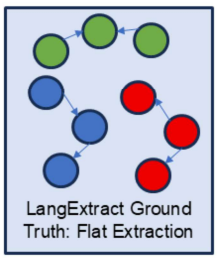
"Second, **Qualitative Validation**: I will detail the 'Controlled Error Injection' experiment."

"This is where I intentionally broke the laws—inserting specific errors into the text—to prove that Mnemosyne could detect the inconsistencies."

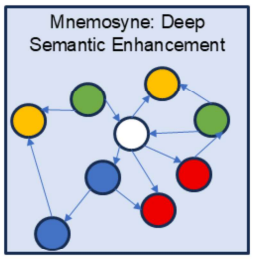
Conclusion: "These experiments provide the empirical backing for the system's capabilities."

Introduction Foundation **Key Findings** Discussion Conclusion

Semantic Depth vs Standard Extraction



Ground Truth Source	Avg. Entity F1	Avg. Relationship F1	Avg. Overall F1
Manual Curation	71.19%	12.09%	41.64%
Google LangExtract	32.53%	2.70%	17.62%



Mnemosyne provided acceptable F1 scores against the manual ground truth for the four hand crafted documents. It did not do as well with the LangExtract created ground truth. Manual observation showed that a large part of the discrepancy was that Mnemosyne created a rich semantic network. Whereas both the manual and LangExtract ground truths were flatter. Low F1 scores indicate Mnemosyne detected relationships the Ground Truth did not include, not an error.

THE GEORGE WASHINGTON UNIVERSITY WASHINGTON, DC

Transition: "I want to address the 'Elephant in the Room' regarding these metrics."

Notes:

"If you look at the table, you see a stark difference: **Manual Curation** (Manually Generated Ground Truth) shows high accuracy at roughly 71% Entity F1."

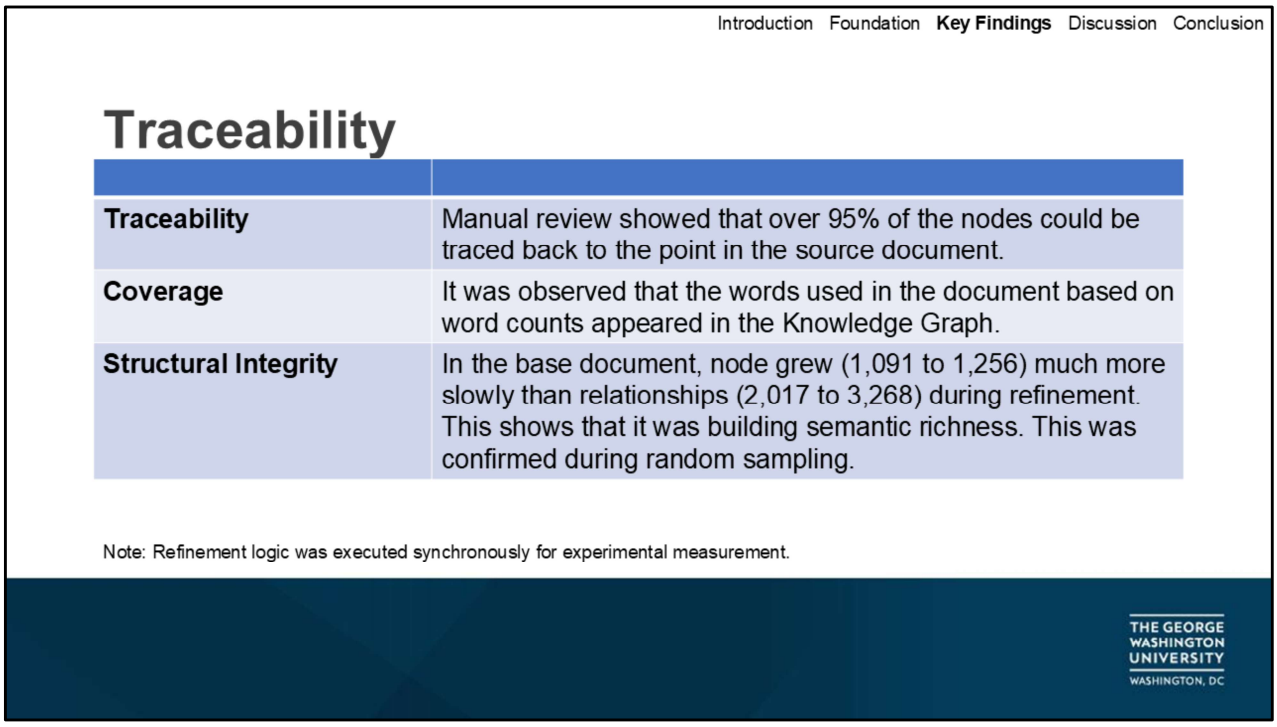
"However, **Google LangExtract** (Machine Generated Ground Truth) shows a much lower entity F1 of roughly 33%."

"In a standard ML context, that 33% would be a failure. Here, it is a measurement of **Semantic Enrichment**."

"LangExtract creates a 'Flat Graph'—it only captures syntax. Mnemosyne creates a 'Deep Graph'—it captures implied semantics. If the text says 'Car,' Mnemosyne infers 'Vehicle.' LangExtract ground truth does not have 'Vehicle' so it is seen as an error because the word wasn't in the sentence."

"Relationship F1 scores are low because there is a greater variety in the terminology used for relationships."

Conclusion: "Therefore, the low F1 score against LangExtract actually validates Hypothesis 2: It proves the system is generating knowledge that strictly exceeds the raw text."



Transition: "So, if the standard F1 scores don't tell the whole story, how do we know the graph is actually fit for purpose? We evaluated the system on three critical dimensions."

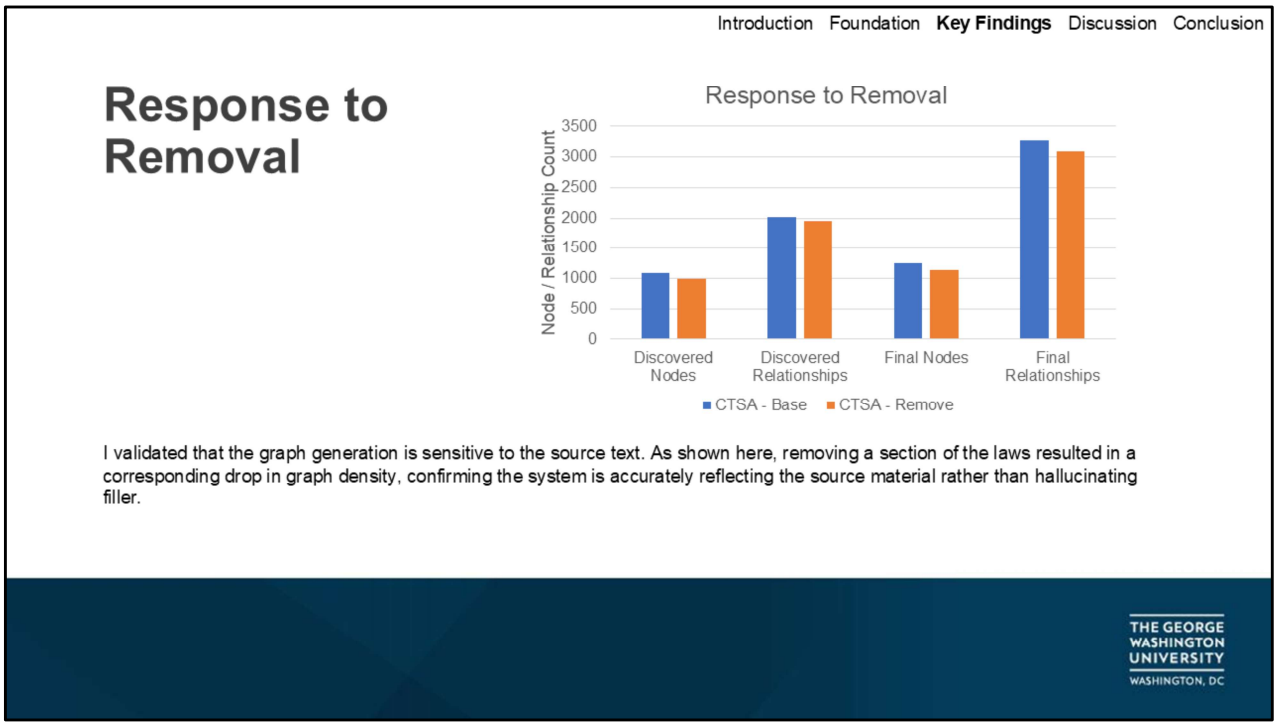
Notes:

"First, **Traceability (Trust)**: One of the biggest risks with Generative AI is hallucination. I found that over 95% of the nodes could be explicitly traced back to a specific location in the source document."

"Second, **Coverage**: I compared word frequency in the source against concepts in the Graph to ensure I wasn't losing the core message."

"Finally, **Structural Integrity**: This is the proof of Semantic Enrichment. During the refinement phase, Nodes grew slowly (about 15%), but Relationships grew explosively (over 60%)."

Conclusion: "This validates my hypothesis: The system wasn't just adding more 'nouns' or random noise; it was aggressively knitting connections between existing concepts to turn a list of facts into a queryable web."



Transition: "While I proved the system can add value through semantic enrichment, I also had to prove it doesn't hallucinate value where none exists."

Notes:

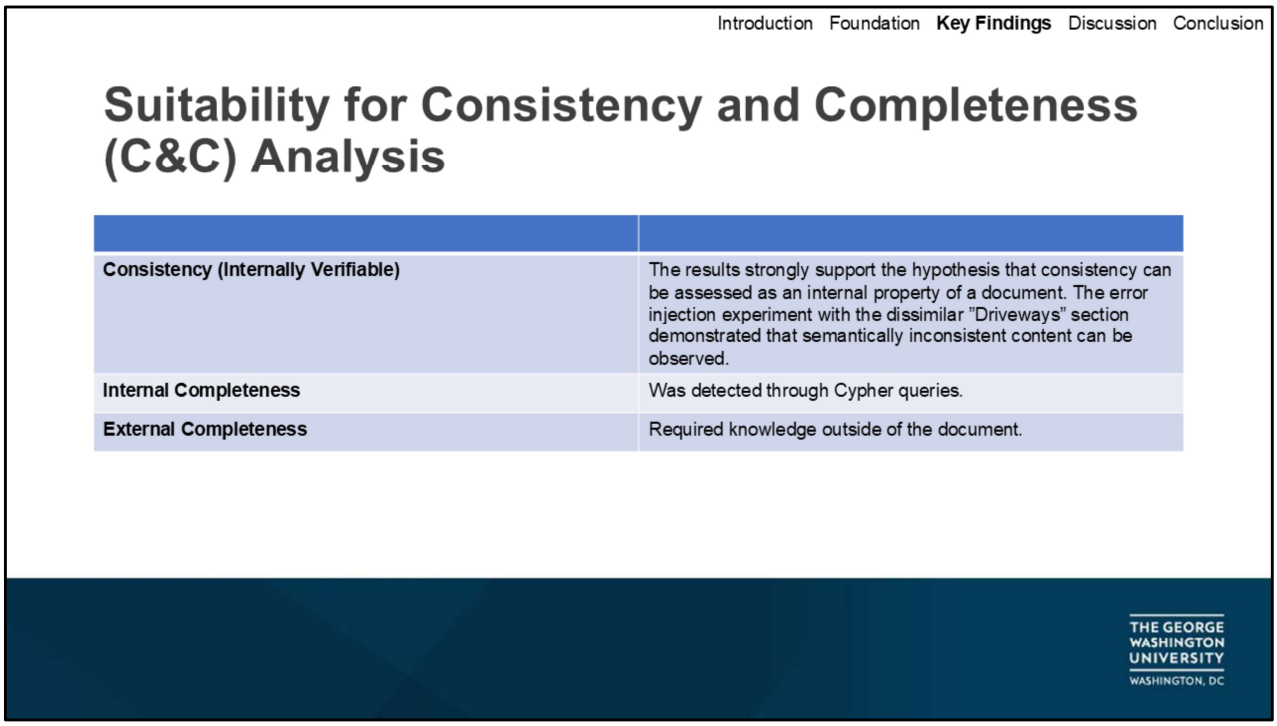
"A common failure mode in LLMs is 'filling in the blank.' To test this, I performed a subtraction experiment labeled here as **CTSA-Remove**."

"I physically deleted a section of the text. The Blue bars represent the full 'Base' document; the Orange bars represent the document with the text removed."

"If the system were hallucinating, I would expect the Orange bars to stay roughly the same height as the LLM tried to compensate. Instead, we see a clear, proportional drop across every metric."

"Final Nodes dropped from roughly 1,250 to 1,130, with a corresponding decrease in relationships."

Conclusion: "This proves Mnemosyne is structurally sensitive to the source text. When the text disappears, the knowledge disappears. It does not invent filler.“



Transition: "Having validated the structure of the graph, I then tested if it could actually answer the core research questions regarding Consistency and Completeness."

Notes:

"First, I looked at **Consistency**. This is verifiable internally. To prove this, I performed a 'Controlled Error Injection' experiment called the 'Driveways' test."

"I injected semantically dissimilar text about driveways into the sewer law corpus. The hypothesis was that if the system were just 'hallucinating connections,' it would blend this in. Instead, the system isolated it as a structural anomaly."

"Second is **Internal Completeness**. This asks: 'Does the document support itself?' By querying for nodes that were used but lacked a DEFINES relationship, I identified gaps strictly using the graph's topology."

"Finally, **External Completeness**. This asks: 'Is the document missing something required by the outside world?'."

Conclusion: "As noted on the slide, External Completeness requires knowledge outside the source document. Mnemosyne can only flag these if the external regulations are also loaded into the graph."

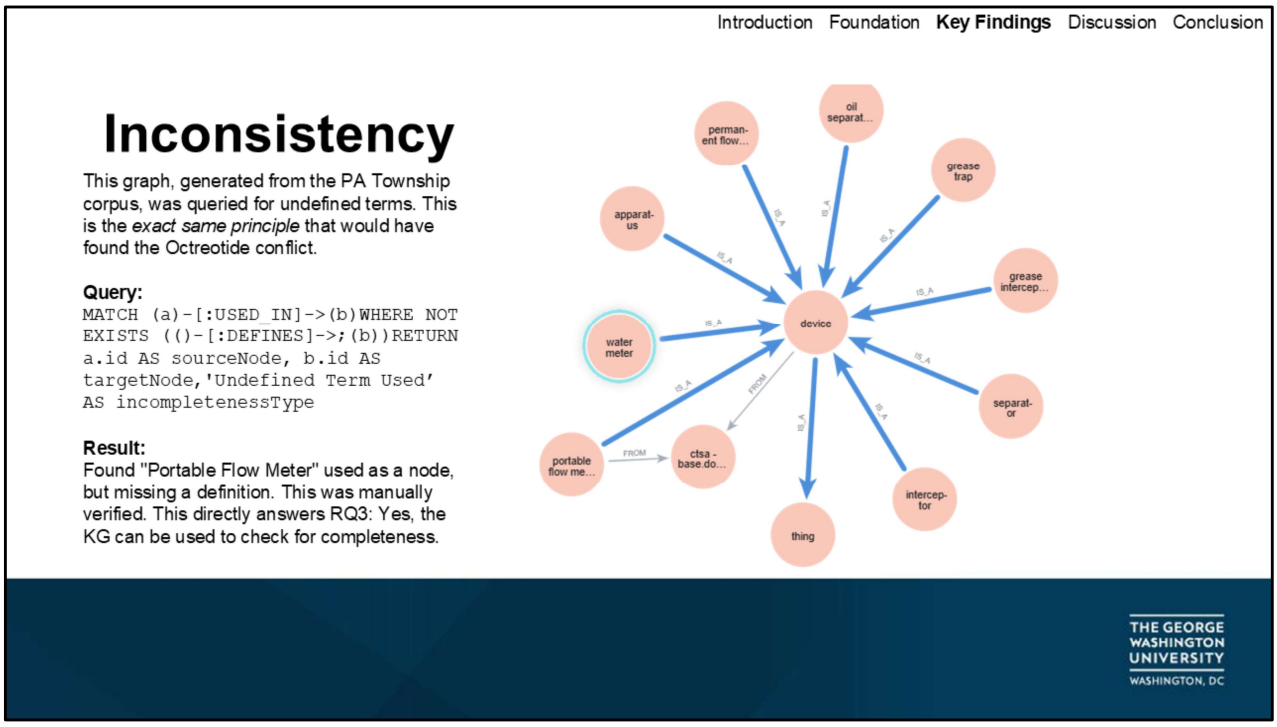
Transition: "I want to draw your attention to the first bullet regarding **Consistency**. While the 'Driveway' experiment successfully proved we can detect *dissimilar* content through topological isolation, I also wanted to find the breaking point of

this method."

The Limitation (The "Sewer" Test): "I performed a second stress test where I injected a 'Prohibited Wastes' section from a neighboring township. Because this injected text shared the exact same domain vocabulary as the source document, the semantic clustering algorithms could not isolate it as an anomaly."

The Insight/Defense: "This is a critical finding. It demonstrates that **semantic analysis** has a 'resolution limit.' It is excellent for catching cross-domain logical errors—like the Driveway or the Octreotide example—but for distinguishing between two nearly identical legal codes, semantic analysis is insufficient.

This confirms that future iterations must integrate **structural metadata**—such as section numbering and header formatting—alongside the semantic graph to catch these 'near-peer' inconsistencies."



Transition: "Finally, we arrive at the core proof of concept: The Query."

Notes:

"On the left, you see the actual Cypher query I ran. The logic is simple: It asks the system to MATCH every instance where a term is USED_IN a regulation, then filters for cases WHERE NOT EXISTS a corresponding node that DEFINES that term."

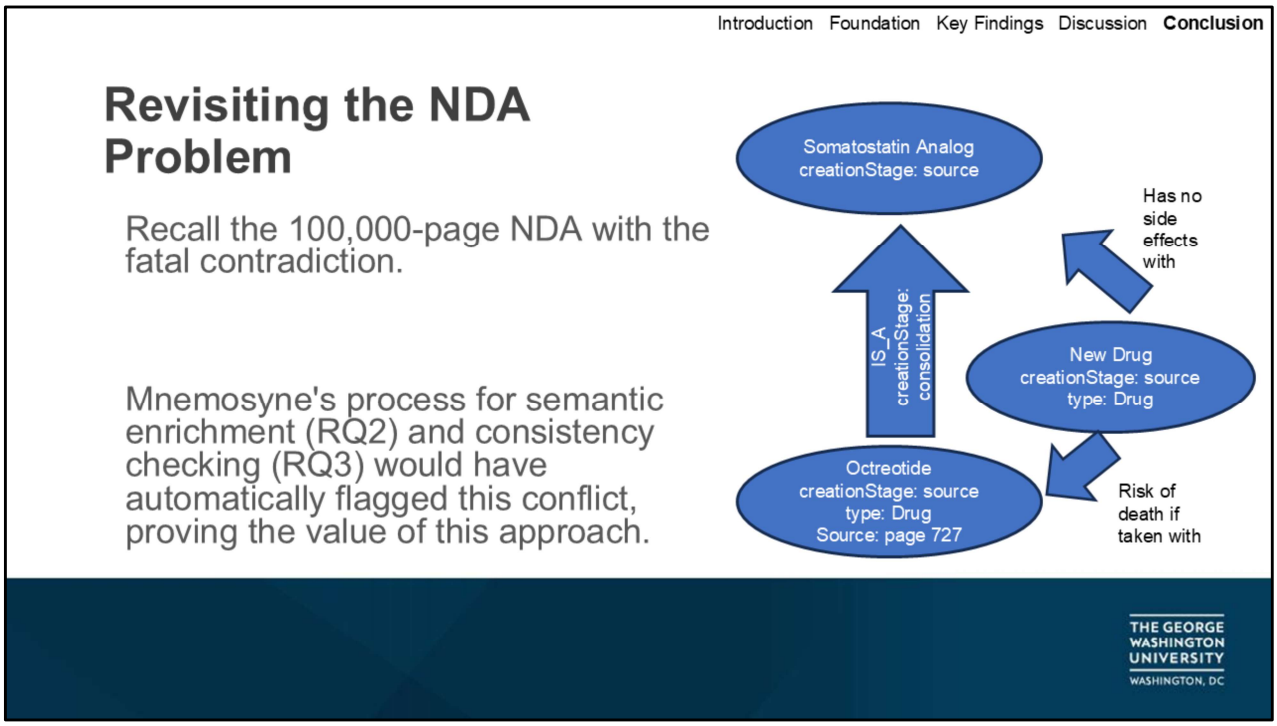
"When running this against the Township Laws, the system instantly flagged the **'Portable Flow Meter'**."

"The graph revealed that the specific composite concept of a 'Portable Flow Meter' was used but never defined."

"This confirms Mnemosyne can detect 'Internal Completeness' failures—identifying concepts that exist in the regulations but are missing their foundational definitions."

"This matters because it is the exact same logical pathway required to catch the 'Octreotide' error in an NDA. It traces non-local logical dependencies across the graph."

Conclusion: "This directly answers Research Question 3: Yes, the semantically enriched graph can automatically detect completeness failures that a linear reading might miss."



Transition: "Now, why does finding a missing definition for a 'Flow Meter' matter? Because the exact same logic used to find that error solves the multi-billion dollar 'Fatal Contradiction' we discussed in the introduction."

Notes:

"Let's apply Mnemosyne's graph structure to that 100,000-page NDA."

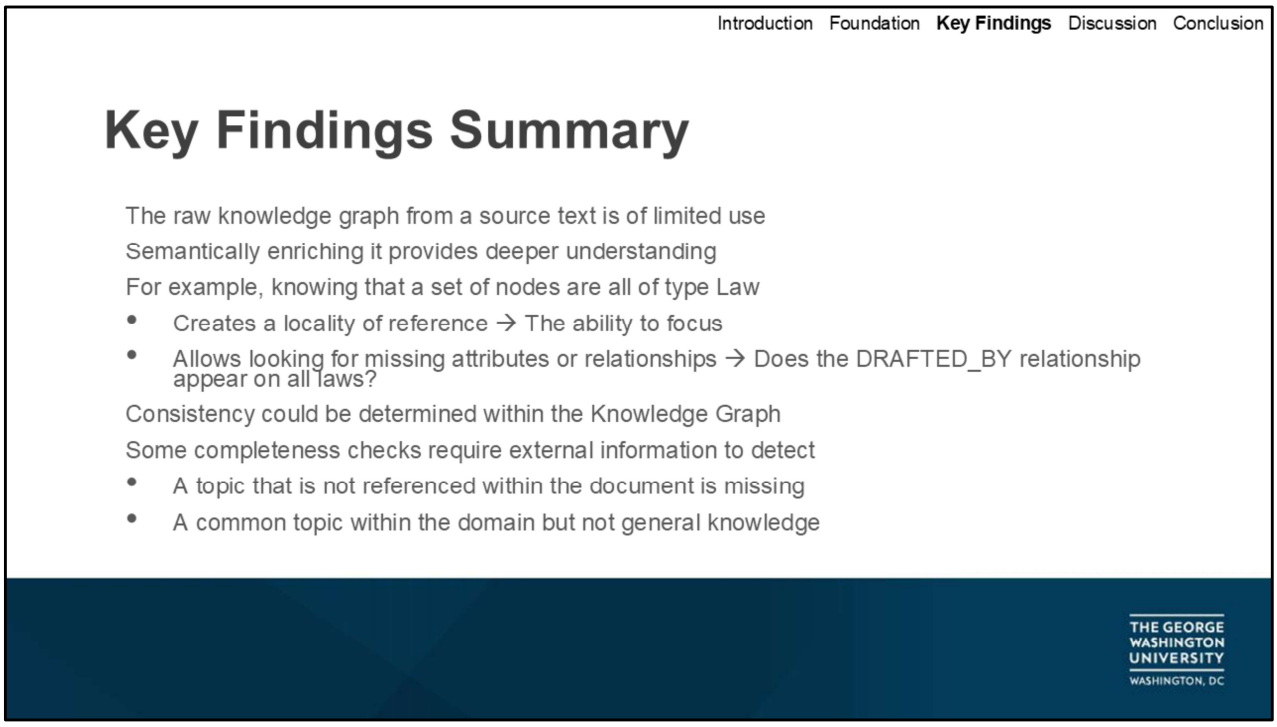
"**Fact A:** The system extracts the specific warning that 'Octreotide' carries a risk of death."

"**Fact B:** Thirty thousand pages later, it extracts the broad claim that 'Somatostatin Analogs' have no side effects."

"**The Bridge:** Crucially, during the Refinement Phase, the system infers the ontological link: that 'Octreotide IS_A Somatostatin Analog'. Note that its creationStage is consolidation."

"Because the graph is now semantically connected, the Consistency Check triggers automatically. It asks: 'Does a specific instance (Octreotide) have a constraint that contradicts its parent class?' The answer is Yes."

Conclusion: "This turns a multi-billion dollar failure into a simple database query. It proves that by converting documents to knowledge graphs, we find the needle in the haystack not by searching for it, but by using a magnet."



Transition: "Based on these experiments—both the real-world Township tests and the NDA logic models—we can distill our findings into three core insights."

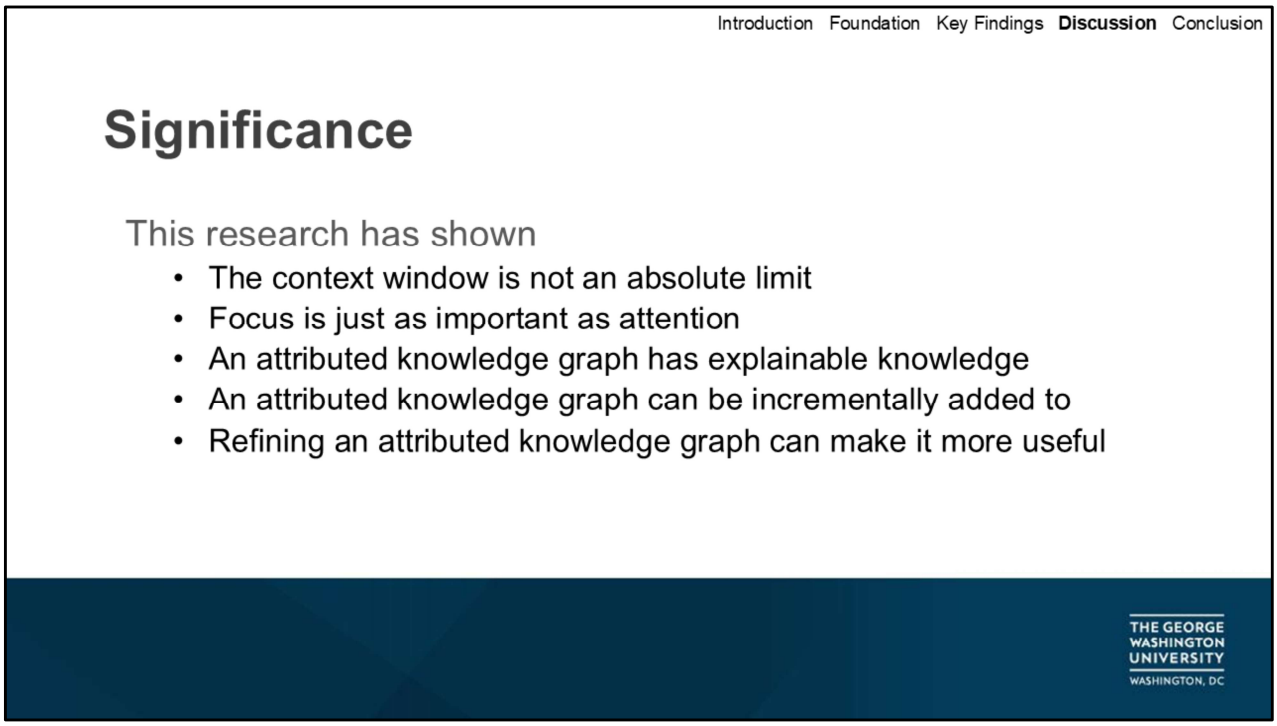
Notes:

"First, I learned that a raw knowledge graph is of limited utility. The real breakthrough comes from **Semantic Enrichment**. By clustering nodes into types (like linking Octreotide to Drug Class), I create a 'Locality of Reference'."

"Second, this clustering enables **Structural Anomaly Detection**. I don't just look for keywords; I look for broken patterns—like a law missing a DRAFTED_BY relationship or a Drug missing a SIDE_EFFECT relationship."

"Third, I established a clear boundary for **Completeness**. Internal Consistency is fully solvable within the document. External Completeness, however, is only solvable if I load the external regulations into the graph as well."

Conclusion: "Mnemosyne strictly adheres to the source text: it creates a closed-world assumption where consistency is guaranteed within the boundaries of the document."



Transition: "These findings have implications that go beyond just error checking. They fundamentally change how to view the constraints of Large Language Models 'Context Window Problem'."

Notes:

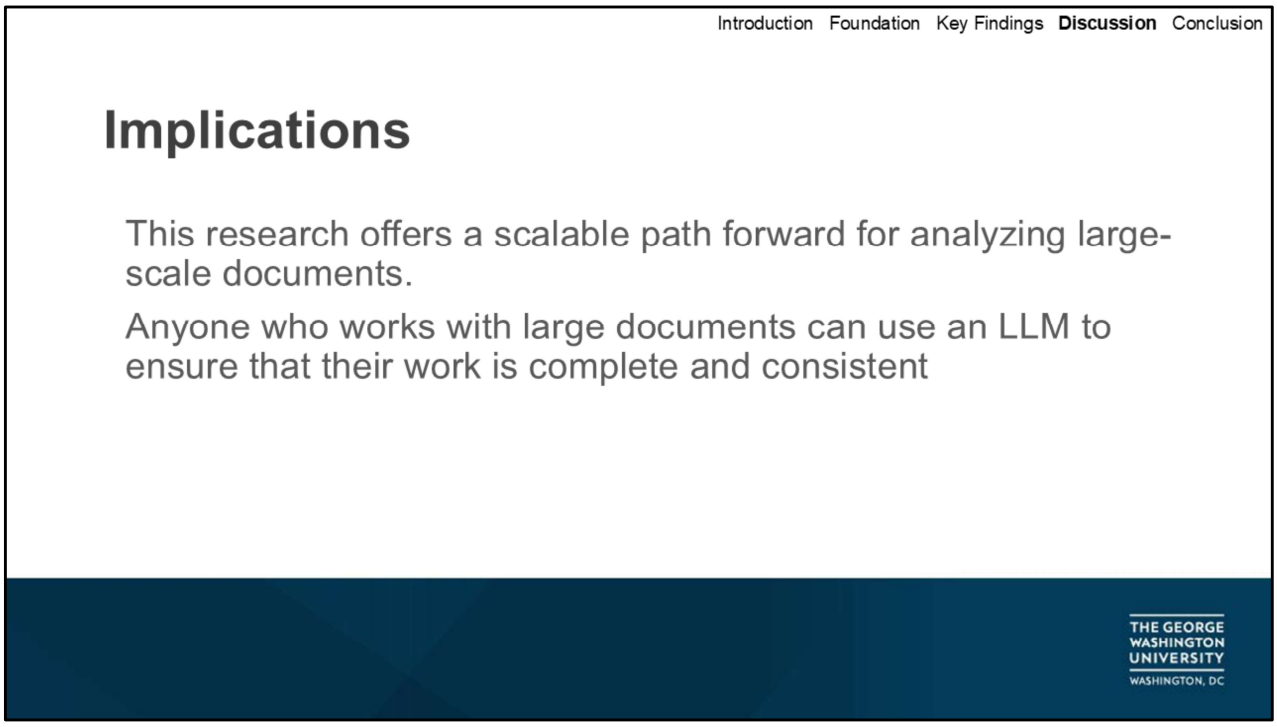
"First, I demonstrated that the **Context Window is not an absolute limit**. By converting text into structure, I decouple 'Knowledge' from 'Memory'."

"Second, I distinguish between **'Attention' and 'Focus'**. Attention is a mechanical property; Focus is the ability to retrieve only the relevant subgraph, allowing the LLM to reason without being distracted by 90,000 pages of noise."

"Third is **Explainability**. A standard LLM is a black box. An Attributed Knowledge Graph is transparent—every node and relationship is an audit trail, which is a requirement in regulated industries."

"Finally, **Incremental Learning**. Because Mnemosyne stores knowledge as a Graph, adding a new amendment doesn't require re-reading the old laws; you simply graft the new nodes onto the existing structure."

Conclusion: "This makes the system scalable in a way that raw context windows will never be."



Transition: "So, what is the practical impact of this architecture? What does it mean for the future of work?"

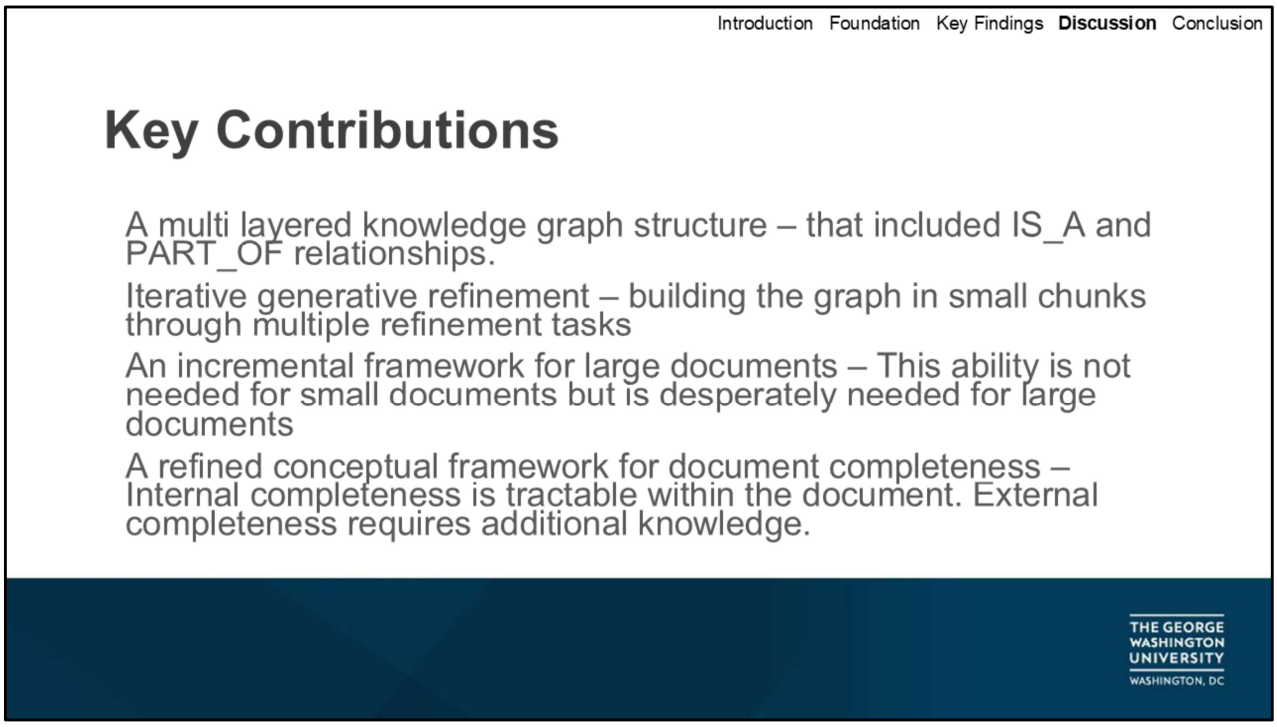
Notes:

"We are currently facing a **Scalability Crisis**. Documents like NDAs have grown beyond human memory and standard LLM capacities. Mnemosyne offers a third path: Scalable Analysis that decouples document size from review quality."

"Crucially, this changes the role of AI from **Generation** to **Assurance**. Most of the world uses LLMs to write code or emails. My research demonstrates their highest value may be acting as a logical backstop for high-stakes documentation."

"This implies a future where the AI acts as an **'Auditor'**. It allows a human expert to ask, 'Did I contradict myself on page 500?' or 'Did I define every term I used?'."

Conclusion: "This doesn't replace the human expert; it simply gives them a tool that has infinite memory and zero attention drift."



Transition: "To formalize the academic output of this study, I have identified four specific contributions to the field of Knowledge Engineering."

Notes:

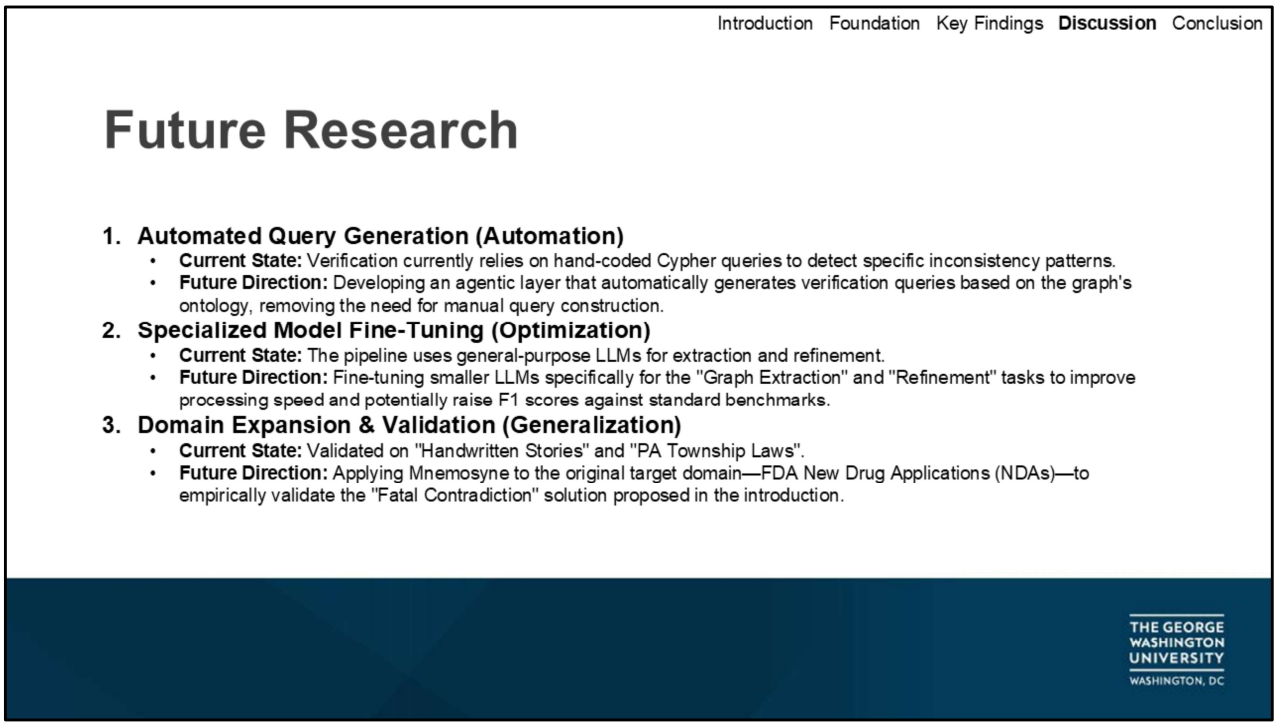
"First is the **Multi-Layered Knowledge Graph Structure**. Previous approaches created 'flat' maps of sentences. My contribution was enforcing a hierarchical ontology using IS_A and PART_OF relationships, transforming the graph into a navigable structure of concepts."

"Second is **Iterative Generative Refinement**. Most extraction pipelines are linear. Mnemosyne introduces a cyclical loop where the system revisits the graph to discover connections that weren't explicit in any single sentence."

"Third is the **Incremental Framework**. I proved I can build the graph in small chunks and merge them, creating a pipeline that is theoretically boundless in terms of document length."

"Finally, a **Taxonomy of Completeness**. I formally separated 'Completeness' into Internal (solvable by the graph) and External (requiring outside data), allowing future researchers to focus on the tractable problem of Internal Consistency."

Conclusion: "These contributions provide the necessary architectural shift to move beyond simple summarization."



Transition: "While Mnemosyne has successfully demonstrated the core concept, there are three distinct avenues for future research to transition this from an academic prototype to an industrial solution."

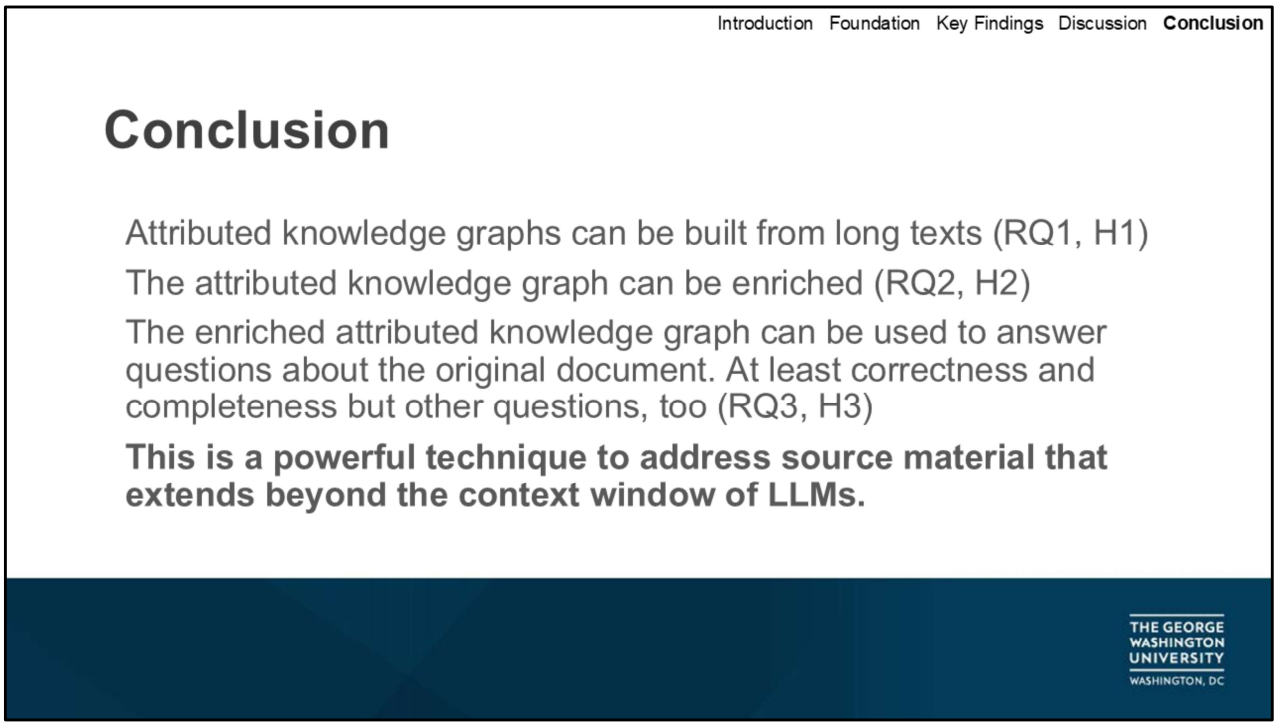
Notes:

"First is **Automated Query Generation** (Automation). Currently, verification requires hand-coded Cypher queries. The next step is to build a Layer that translates natural language questions—like 'Are there undefined terms?'—into the necessary database queries. The other side is a component that will display the results of the queries."

"Second is **Specialized Model Fine-Tuning** (Optimization). Instead of using expensive general-purpose models, future work should focus on fine-tuning smaller models specifically for entity/relationship extraction and knowledge graph refinement. This would increase speed and likely improve F1 scores by reducing semantic noise."

"Finally, **Domain Expansion** (Generalization). The ultimate goal is to apply Mnemosyne to a real, 100,000-page confidential FDA New Drug Application."

Conclusion: "This would be the empirical gold standard, proving definitively that the 'Fatal Contradiction' scenario can be prevented in a live regulatory environment."



Transition: "In conclusion, I return to the three core hypotheses that drove this research. The results allow me to answer each in the affirmative."

Notes:

"First, regarding **RQ1 & H1**: I confirmed that an Attributed Knowledge Graph can be successfully constructed from massive, unstructured texts without losing the fidelity of individual entities."

"Second, regarding **RQ2 & H2**: I proved that this graph can be Enriched. The 'Refinement Phase' demonstrated that I can move beyond keyword extraction to build a semantic layer that functions as 'Long Term Memory'."

"Third, regarding **RQ3 & H3**: I demonstrated Utility. As shown with the 'Portable Flow Meter' discovery, the system can successfully verify the correctness and completeness of a source document in ways that linear reading cannot."

Conclusion: "Ultimately, Mnemosyne proves that the solution to the 'Context Window' limit is not making the window bigger—it is about changing how we view the data, building a system that can 'navigate' rather than just 'remember'."

Transition: “Thank you for the opportunity to present my research today. Are there any questions?"

Notes:

“Answer questions."

Conclusion: “Thank you."

Likely questions:

Likely Question 1: The Recursion Problem

The Question: "You claim to solve the context window limit, but your 'Refinement' step sends chunks to the LLM. Aren't you just moving the token limit problem to a different step?"

The Answer: "That is a fair point, but the distinction is in *what* I am sending. I am not sending the whole document to be refined at once. I send small, random sub-graphs. Because the 'Long Term Memory' (Neo4j) acts as the persistent global state, these small, local updates accumulate into a globally consistent graph. It’s similar to how a painter focuses on one corner of a canvas at a time; they don't need to hold the entire image in their focal point to make the whole painting consistent.“

Likely Question 2: The F1 Score

The Question: "Your F1 scores against Google LangExtract are near zero. In any other defense, this would be a failure. Why should we accept this?"

The Answer: "If my goal were to build a better *summarizer*, a low F1 score would be a failure. But my goal was **Semantic Enrichment**. The F1 score measures 'Word Overlap.' If the text says 'Car' and my graph says 'Vehicle,' the F1 score marks that as an error. But ontologically, that is an *insight*, not an error. My manual review of the handwritten stories confirmed that when a human judges the output, the accuracy jumps to over 70%. The low F1 score against LangExtract is actually a quantitative measure of how much *new* semantic structure Mnemosyne added that wasn't in the raw text.“

Likely Question 3: Completeness Boundaries

The Question: "You talk about completeness, but you admit you can't find missing external regulations. Isn't that a fatal flaw for an NDA?"

The Answer: "It is a limitation, but not a fatal one. It is a scoping definition. Mnemosyne solves **Internal Completeness**—checking if the document agrees with itself (e.g., defining terms it uses). **External Completeness** is solved by simply loading the external regulations into the graph as a second document. The architecture supports it; it just wasn't the focus of this specific experiment.“

Likely Question 4: Hallucination

The Question: "How do we know the system isn't just making up these relationships? How do you prevent hallucination in a safety-critical graph?"

The Answer: "We tested for exactly this using the 'Removal Experiment' (Slide 18). When I removed text from the source, the corresponding nodes in the graph disappeared. The system did not try to 'fill in the gap' with guessed information. Furthermore, because every node in the graph is tagged with a source attribute (traceability), a human auditor can instantly verify where a fact came from. We prioritize **Traceability** over **Creativity**.“

Likely Question 5: The "Semantic Hallucination" Trap

The Question: "On Slide 15, you defend an 8% F1 score by calling it 'Semantic Enrichment.' But in any other engineering discipline, a 92% deviation from the ground truth is called 'Noise' or 'Hallucination.' If the graph doesn't match the text, how can you legally certify an NDA based on it? Aren't you just introducing *more* risk?"

The Answer: "I agree that for *summarization*, 8% is a failure. But for *inference*, strict adherence to syntax is a bug, not a feature.

- **The Evidence:** Look at the **Manual Curation** score. When a human expert created the ground truth, the accuracy jumped to **71%**.
- **The Logic:** The 92% 'error' marked by LangExtract was mostly the system adding IS_A and PART_OF types. An NDA checker *needs* to know that 'Octreotide' is a 'Somatostatin Analog' to find the contradiction. A 100% F1 score would mean

the system learned nothing new, rendering it useless for this specific task.“

Likely Question 6: The "Driveway" Instability

The Question: "Look at Slide 35. When you added the 'Driveway' text, your total node count **dropped** from 1,256 (Base) to 1,228. You added text, but you lost information. In a Pharma context, if I add a safety warning and the system accidentally drops a contraindication elsewhere because of 'noise,' people die. How do you justify this instability?"

The Answer: "This is the trade-off of using probabilistic models (LLMs) for extraction.

The Cause: The drop of ~20 nodes represents less than a 2% variance. This happened because the 'Driveway' text acted as semantic noise, slightly diluting the attention mechanism during the initial extraction pass.

The Mitigation: In a production environment, we would run the extraction multiple times and merge the results (Ensemble Extraction) to eliminate this stochastic variance. However, even with this drop, the *structural* integrity of the Sewer Laws remained intact. The system didn't hallucinate false connections; it just became slightly more conservative.“

Likely Question 7: The "Obsolete Research" Trap

The Question: "Since you started this research, Gemini and GPT-4 have released context windows of 1 million+ tokens. You can now fit the entire PA Law corpus into a prompt. Why build a complex, expensive Graph pipeline when I can just upload the PDF and ask: 'Are there contradictions?' Is your research already obsolete?"

The Answer: "A larger window does not equal better reasoning; it just means more reading. This is the difference between **Recall** and **Reasoning**.

The Problem: Research shows that as context fills up, 'Attention Drift' occurs—the model starts to ignore the middle of the document (the 'Lost in the Middle' phenomenon).

The Solution: Mnemosyne doesn't just *store* the text; it *structures* it. By converting the document to a Graph, we turn a probabilistic 'Search' problem into a deterministic 'Lookup' problem. No matter how big context windows get, they cannot beat the auditability and precision of a Cypher query for finding a specific logical fault.“

Likely Question 8: The "Black Box" Liability

The Question: "You used 'Refinement' to have the LLM looking at the graph and creating new relationships. If the LLM incorrectly links 'Drug A' to 'Class B' during this creative phase, and the graph saves it to Long Term Memory, you have permanently encoded an error. How does a user verify the graph is correct without re-reading the whole document themselves?"

The Answer: "This is why **Traceability** was a core metric.

The Safety Net: Every node Mnemosyne generates is tagged with creationStage and source attributes, linking back to the specific paragraph.

The Workflow: The system is not an 'Autopilot'; it is a 'Copilot.' When the graph

flags a contradiction, it provides the two source nodes. The human reviewer doesn't have to read the whole document—they only have to click the link to verify those two specific nodes. We reduce the workload from 100,000 pages to 2 paragraphs."


THE GEORGE
WASHINGTON
UNIVERSITY
WASHINGTON, DC

Appendix

Details

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Knowledge Graph Fit Assessment

Experiment Document	Overall F1 (%)	Entity F1 (%)	Relationship F1 (%)
CTSA Base Document	7.95	15.90	0.00
CTSA with Content Removed	7.85	15.33	0.36
CTSA with Sewer Content Added	7.90	15.79	0.00
CTSA with Driveway Content Added	8.64	17.07	0.20

Values are low because LangExtract had to be used to create the ground truth and its flat graph does not match well with the semantically rich graph from Mnemosyne.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

The Setup (Acknowledge the numbers): "I anticipated this question. If you look at the table here, you see F1 scores hovering between 7% and 16% across all experimental conditions. I want to be direct: if my goal were simple text summarization or keyword extraction, these numbers would indeed be failures."

The Pivot (Redefine the metric): "However, the low score here is not a failure of extraction; it is a measurement of **Semantic Enrichment**. The Ground Truth for this table was generated by **Google LangExtract**, which creates a 'Flat Graph.' It only captures what is explicitly written. Mnemosyne, by design, creates a 'Deep Graph.' It infers the ontology."

The Concrete Example (The "Sewer" Defense): "Let me give you a specific example from this data. If the text says 'The Sewer Authority shall manage waste,' LangExtract extracts 'Sewer Authority' and stops. Mnemosyne extracts 'Sewer Authority,' but then adds a node classifying it as an **'Organization'** via a consolidated relationship. Because the word 'Organization' was not in the source sentence, the F1 metric penalizes Mnemosyne for a 'False Positive'. Therefore, the low F1 score is actually quantifying the **new knowledge** the system added that the standard extractor missed."

The Proof (The "Human" Standard): "To prove this wasn't just 'hallucination,' we refer back to the **Manual Curation** experiment I showed earlier. When a human expert created the ground truth, the Entity F1 score jumped to **71.19%**. This confirms that

when judged by intelligence rather than syntax, the system performs at a high level."

Key Findings

Document	Total Paragraph Count	Discovered Nodes	Discovered Relationships	Final Nodes	Final Relationships
NER 1	21	67	133	84	228
NER 2	20	63	124	90	233
NER 3	20	64	131	89	241
NER 4	33	104	205	127	352
CTSA – Base	1,259	1,091	2,017	1,256	3,268
CTSA – Remove	No EDA	986	1,955	1,133	3,099
CTSA – Add Sewer	No EDA	1.095	2,115	1,256	3,366
CTSA – Add Driveway	No EDA	1,072	1,884	1,228	3,098
ET – Base	13,951	No Run	No Run	No Run	No Run

Sources:
RUN_2025-09-10_01-51-12
RUN_2025-09-19_01-48-36

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Entry/Navigation Script: *(Use if they ask about data stability or specific node counts)*

"I have the raw data table here that breaks down the exact node and relationship counts for every experimental run."

The "Driveway Anomaly" Defense: *(Use if they ask: "Why did the node count drop when you added the Driveway text?")*

1. The Direct Answer (Semantic Dilution)

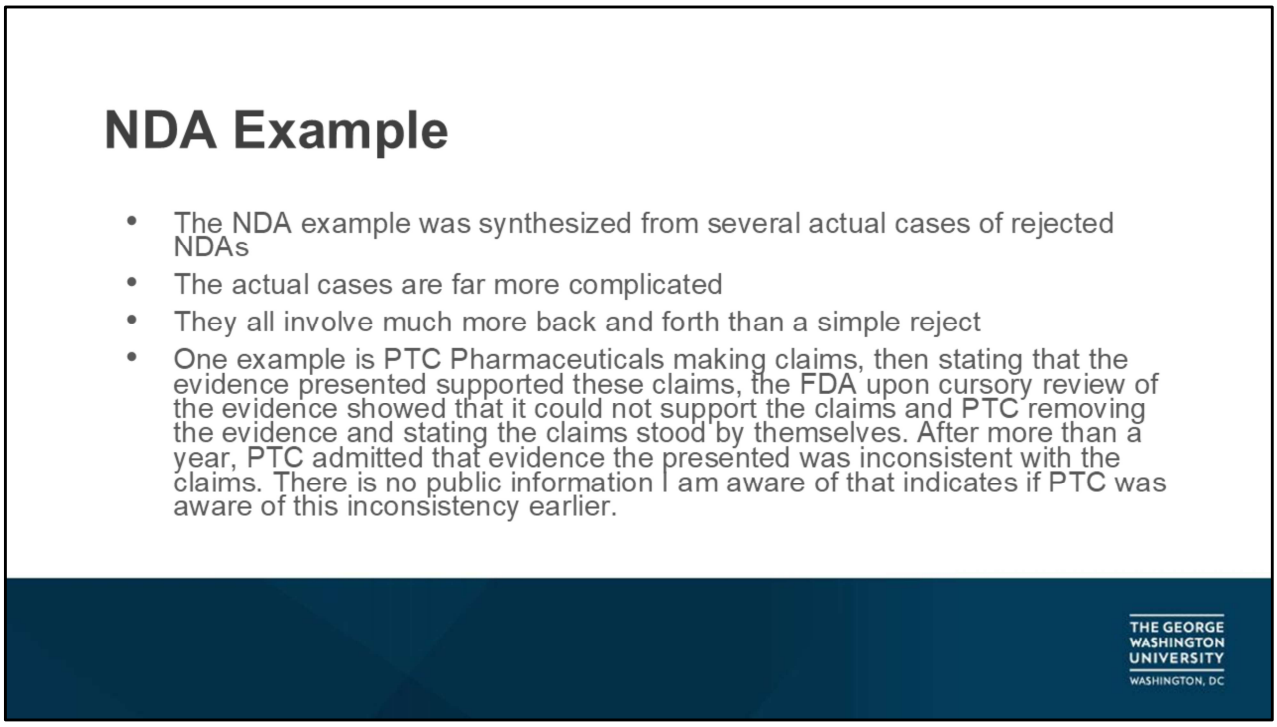
"That is a sharp observation. You are looking at the drop from 1,256 nodes in the **Base** run to 1,228 nodes in the **Driveway** run. This happens due to **Semantic Dilution**. The 'Driveway' text was intentionally chosen because it is semantically dissimilar to the 'Sewer' domain. When we introduce unrelated concepts, they act as noise during the extraction phase. This noise slightly dilutes the attention mechanism, causing the model to miss a few of the weaker, marginal signals it caught in the clean run."

2. The Evidence (Point to the 'Discovered' Column)

"If you look at the **Discovered Nodes** column (the third column), you will see the count dropped from 1,091 to 1,072. This proves the loss happened at the very beginning—during ingestion—and not just during refinement. The model effectively deprioritized 'fringe' information because the overall coherence of the document decreased."

3. The Defense (Stability)

"However, the variance is only about 2% (28 nodes). The important takeaway here is what *didn't* happen: The graph didn't explode with irrelevant nodes. The system successfully identified the 'Driveway' text as an outlier and mostly treated it as background noise, which validates the stability of the ontology."



Entry/Navigation Script: *(Use if they question the realism of the Octreotide example)*

"That is a fair critique. While the Octreotide narrative was simplified for the presentation, the logical failures it represents are drawn directly from recent FDA rejection letters."

The Setup (The "Composite" Defense):

"I synthesized the Octreotide case because real NDAs are confidential. However, the specific failure mode—logical inconsistency between claims and evidence—is documented in the public record."

The "Smoking Gun" Example (PTC Therapeutics): *(Focus on this one—it is the strongest parallel to your research)*

"The strongest parallel to my research is **PTC Therapeutics**.

The Situation: They made specific claims in their application and originally pointed to evidence to support them.

The Contradiction: Upon review, the FDA found the evidence didn't support the claims. PTC removed the evidence but *left the claims in the document*.

The Result: This created a 'floating claim'—conceptually, a node in a graph with no supporting edges. It took over a year to resolve this inconsistency. Mnemosyne would have caught this immediately as an 'Orphan Node'—a claim that exists without a valid SUPPORTED_BY relationship."

The "Scope" Example (Zealand Pharma): *(Use this if they ask about External*

Completeness)

"We also see cases like **Zealand Pharma**. Their drug was rejected not because of the text, but because of a third-party manufacturing failure. This validates my point about 'External Completeness'—some failures require knowledge that exists outside the document boundaries."

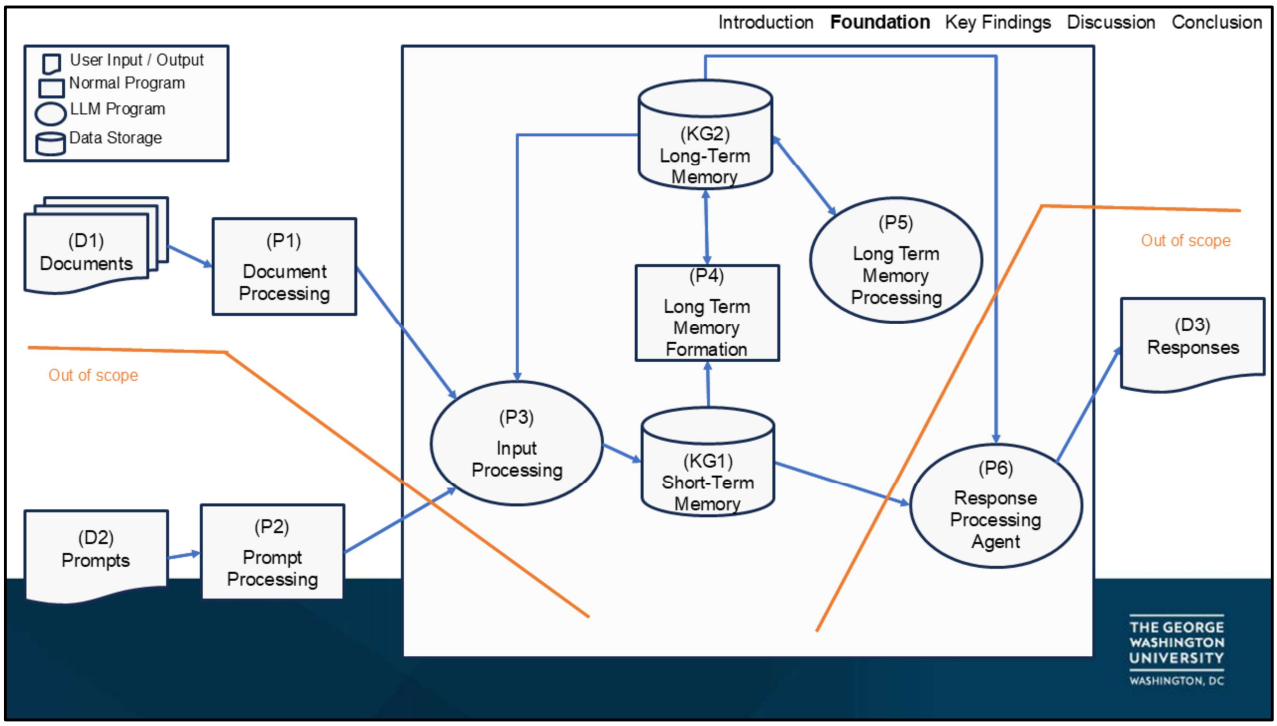
NDA Example

Vanda Pharmaceuticals: The FDA rejected Vanda's proposed dissolution specifications for both its drugs *Fanapt* and *Hetlioz* based on the data provided, requiring the company to adopt the FDA's alternative specification. In another instance, the FDA raised concerns about the "interpretability" of study data when rejecting a supplemental NDA for pimavanserin due to unclear clinical evidence.

PTC Therapeutics: The FDA issued a Refusal to File letter for *Translarna* because both Phase 3 studies failed to meet their primary endpoints, suggesting insufficient and likely inconsistent data to permit a full review.

Zealand Pharma: The NDA for dasiglucagon was rejected not for clinical data issues, but due to deficiencies found during an inspection at a third-party manufacturer, highlighting the importance of consistent quality across all aspects of the application.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC



Entry/Navigation Script: *(Use if they ask: "How exactly does the data move from the document to the graph?" or "Can you explain the memory architecture?")*

"To handle the context window limit, I couldn't just dump the text into a database. I had to design a strict pipeline that manages the handoff between volatile and persistent memory. This diagram maps that specific data flow."

The "Isolation" Defense (D vs. P):

"The first thing to note is the strict separation between **Data (D)** and **Processing (P)**. * **D-Blocks (Documents):** These are treated as immutable. The system never modifies the source text, ensuring we preserve the original evidence.

P-Blocks (Processing): This is where the logic lives. By keeping them separate, we ensure the pipeline is auditable."

The "Memory Handoff" (The Core Mechanism): *(This is the most important part of the slide)*

"The core innovation here is the handoff between **KG1** and **KG2**.

KG1 (Short Term Memory): This is a lightweight, in-memory JSON buffer. It catches the output from the LLM immediately after chunking. It is volatile—if the system crashes here, the chunk is lost.

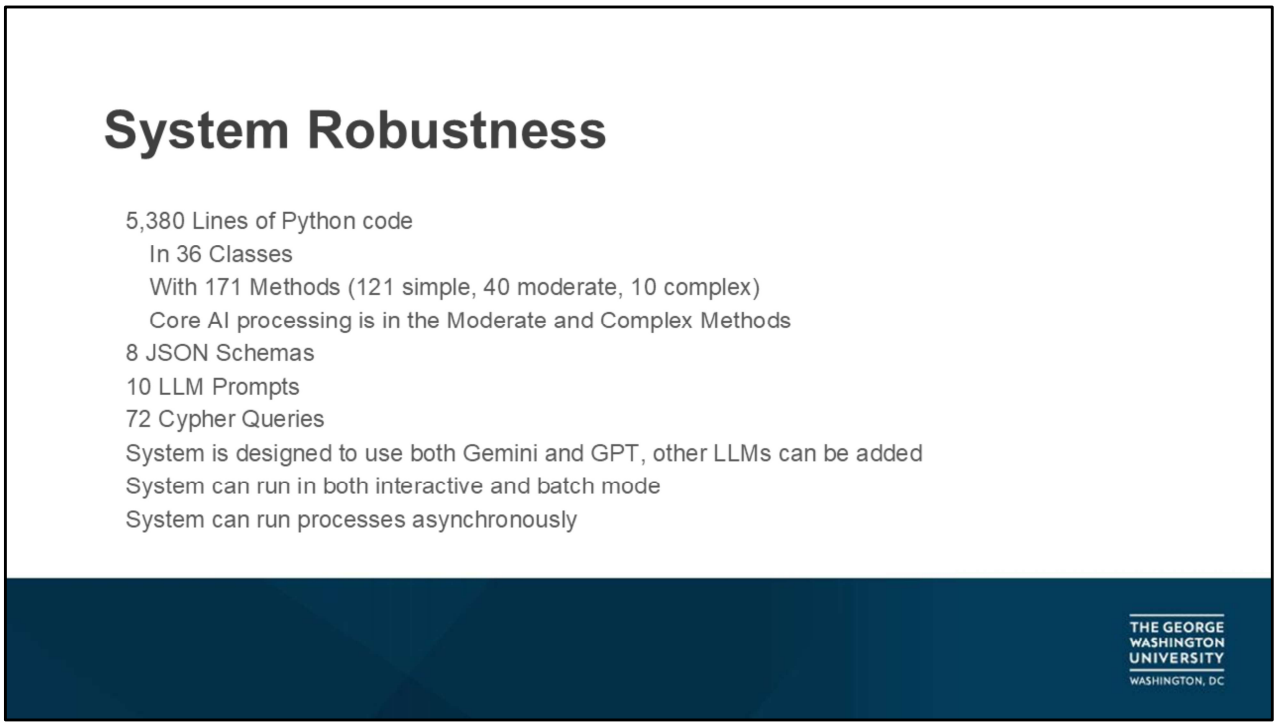
KG2 (Long Term Memory): This is the Neo4j database. Process **P4** is responsible for the 'Consolidation' step—moving data from volatile memory to persistent storage.

Once data crosses into KG2, it is safe and can be refined by the system without needing

to re-read the document."

The "Future Proofing" (The Agent):

"Finally, you will see **P2 (Prompt Processing)** and **P6 (Response Processing Agent)** marked as 'Out of Scope'. In a production application, this is where the chatbot would live—translating user questions into the Cypher queries we manually generated for this research. Then presenting the results back to the user. A key point is that the prompt and response are not directly connected."



Entry/Navigation Script: *(Use if they question the complexity or reproducibility of your code)*

"To ensure the results were reproducible and not just a 'lucky prompt,' I had to build a fairly robust software architecture around the LLM. I have a breakdown of the codebase statistics here."

The "Not Just a Script" Defense:

"The key takeaway from this slide is that Mnemosyne is not just a Jupyter Notebook or a simple script. It is a modular system comprising over **5,000 lines of Python code** across **36 distinct classes**.

I specifically separated the architecture into 'Mind Classes' (which handle the AI logic) and 'Graph Database Classes' (which handle the storage). This modularity was essential for managing the asynchronous calls required to process large documents without timing out."

The "Prompt Engineering" Defense:

"You can also see that the system relies on **72 hand-coded Cypher Queries** and **10 hand-coded LLM Prompts**. These aren't hard-coded strings hidden in functions; they are managed as external configuration assets (JSON Schemas).

This ensures that every run uses the exact same instructions, making the experiment scientifically reproducible."

The "Model Agnostic" Feature:

"Finally, I designed the system to be model-agnostic. While I primarily tested with Gemini and GPT, the LLM Classes abstraction layer allows us to swap in Llama or Claude with minimal code changes. This future-proofs the research against the rapid turnover in the model market."

System Robustness

	Loc	Classes	Methods
Imports and Setup	855	10	1
Graph Database Classes	880	6	38
LLM Classes	287	2	13
Document Classes	445	2	26
Mind Classes	1,117	6	33
Experiment Classes	1,784	10	60
Main Execution	12	-	-
Grand Total	5,380	36	171

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC