

# Organized Thesis Ideas

Here are the organized and refined thesis ideas based on existing notes:

## 1. Attention Mechanisms and Information Retrieval

- **Rectangular Attention for Triples:** Explore using rectangular attention to focus transformer models on specific "triples" within a sequence, ensuring that the triples themselves are not self-attending and the rest of the context is also considered. This concept builds upon [rectangular attention](#).
- **Attention with Subdivisions and Matching:** Develop a project that combines the [attention idea](#) with matching techniques, potentially similar to those used in [RetroLLM](#). This could involve subdividing text and creating triples (fact -> citation, and other combinations). The idea also extends to quantifying counterfactuals to measure credit for past actions, leading to more effective policy gradient methods for limited data.
- **Efficient Search with Data Structures:** Investigate the use of the [Burrows–Wheeler transform](#) and [FM-index](#) for efficient citation generation and context matching within the model. Additionally, explore the [Wavelet Tree](#) for efficient search operations.

## 2. Large Language Models (LLMs) and Agent-Based Systems

- **LLMs as Meta-Agents:** Propose a system where LLMs act as meta-agents, responsible for creating plans, scheduling actions, and either executing them or delegating tasks to other LLMs. This aligns with the concept of [meta agent](#).
- **LLM Workflow for Explanations:** Design an LLM-driven workflow that integrates [Manim](#) for visualizations and [Speech synthesis](#) for audio, to generate comprehensive explanations. This forms the basis of a [manim agent](#).
- **Structured Input Formats:** Define an input format for LLMs where content is primarily media (audio, PDF, images) with accompanying text at the end. This is crucial for [input format](#).
- **Framework for Thesis Development:** Establish a "smarter to smaller" framework for thesis development, focusing on high-level "vibe coding" and then detailing specific instructions and relations, avoiding over-complication in smaller steps. This is outlined in [framework thss](#).
- **Guides as Consistent Scratchpads:** Implement the concept of creating guide files that function as consistent scratchpads for intermediate computations with language models, inspired by [Show Your Work: Scratchpads for Intermediate Computation with Language Models](#).
- **Digital Twins for Intelligent Agents:** Explore the use of digital twins to provide intelligent agents with a mechanism for verifying information and actions, as seen in [small rooms](#).
- **Agent Research Workflow:** Define a precise two-step workflow for agent research:
  1. **Accurate Factual Extraction:** Utilizing methods like Claimify (as seen in [metropolitanskyEffectiveExtractionEvaluation2025](#)), ensure precise factual extraction.
  2. **Consistent Idea Entailing:** Employ techniques like Chain of Thought from [Prompt engineering](#) for coherent idea development. This is a core part of [research agents workflow idea](#).
- **Report Writer Agent:** Develop an agent that can generate text in markdown files within Obsidian and automatically cite sources, functioning as a [report writer](#).
- **Query and Workflow Agent:** Implement a workflow where a model first generates a plan to identify relevant text and relational information from semi-structured data, and then an LLM agent traverses this data to extract the necessary information. This is the basis for a [query and workflow idea](#).

## 3. Multimodal LLMs and Spatial Understanding

- **Spatial Understanding with Gaussian Splatting:** Investigate the integration of [Gaussian splatting](#) within [Large language models](#) to facilitate a form of spatial understanding. This is a key aspect of [spatial understanding](#).
- **Native Image Generation from PDFs:** Explore the use of [Gemini](#)'s [Native Image Generation](#) capabilities to extract and include images from PDF documents into [Surveys](#). This concept is explored in [figures gemini](#).