Project Roadmap 🗺️

Last Updated: [DATE] (After V[XX])

This document provides the strategic vision spanning all experiments. It's organized by **architectural phases** (e.g., Embedding, Memory, Gating) and updated after each experiment to reflect progress and pivots.

Vision & North Star 🌟

High-Level Goal: [Describe the ultimate objective, e.g., "Build a Transformer with infinite context at constant memory"]

Success Criteria (Project-Level):

- ☐ Target Metric 1 (e.g., >90% Recall @ 512k context)
- ☐ Target Metric 2 (e.g., >100k tokens/sec throughput)
- ☐ Target Metric 3 (e.g., <15% gate density)

Current Status: [Brief summary, e.g., "V67 achieved 88% Common Recall but throughput dropped to 55k. V68 aims to recover speed via Lane Unification."]

Phase Breakdown

Phase 1: Foundation & Baseline (V1 - V20)

Objective: Establish basic architecture and prove concept viability.

Key Components:

- Embedding layer
- Base attention mechanism
- Initial memory prototype

Key Experiments:

Version	Goal	Status	Outcome
V1	Vanilla Transformer baseline	✅ Complete	45% recall @ 10k context
V5	Add simple fixed-size memory	✅ Complete	60% recall, but memory doesn't scale
V12	Attention masking fix	✅ Complete	Prevented future token leakage
V20	Diagnostic suite	✅ Complete	Gradient flow confirmed

Learnings:

- Fixed-size memory doesn't scale beyond 10k context
- Gradient flow issues require careful initialization

- Need dynamic memory allocation strategy

Next Phase Trigger: Fixed-size memory proves insufficient → Pivot to sparse recurrent memory

---

Phase 2: Sparse Memory Architecture (V21 - V40)

Objective: Implement scalable sparse memory with dual-lane addressing.

Key Components:

- Recurrent Sparse Memory (RSM)
- Dual-lane addressing (Lexical + Semantic)
- Top-K retrieval

Key Experiments:

Version	Goal	Status	Outcome
V23	RSM prototype	✅ Complete	75% recall @ 50k, but slow (30k tok/s)
V28	Dual-lane addressing	✅ Complete	82% rare recall, 0% common recall
V35	Vectorized RSM	✅ Complete	Throughput boost to 110k tok/s
V40	Integration tests	✅ Complete	Confirmed architecture stability

Learnings:

- Dual-lane addressing enables rare token recall
- Vectorization is critical for throughput
- Common words need different gating logic than rare words

Next Phase Trigger: Rare recall works, but common recall fails → Need intelligent gating

---

Phase 3: Intelligent Gating (V41 - V67)

Objective: Develop gating logic that captures both rare and common tokens based on contextual importance.

Key Components:

- Learned salience (V66)
- Regret gating (V67)
- Stratified Z-score gating (V67.3)

Key Experiments:

Version	Goal	Status	Outcome
V66	Learned salience head	✅ Complete	97% rare, 0% common (collapsed)

Version	Goal	Status	Outcome
V67.2	Regret gating (surprisal)	✅ Complete	88% common, 71% rare, 59k tok/s
V67.3	Stratified gating	✅ Complete	Best balance: 77% rare, 83% common @ 23% density
V67 Strategy B	Read-before-write redundancy suppression	✅ Complete	83% common but 55k tok/s (3x slowdown)

Learnings:

- Learned salience collapses to frequency bias
- Regret gating (surprisal) successfully captures common words
- Stratified Z-scores work but density is high (23%)
- Read-before-write kills throughput

Open Questions:

- Can we hit <15% density without sacrificing recall?
- How to restore throughput to >100k tok/s?

Next Phase Trigger: Gating works but throughput tanks + density too high → Unify lanes to reduce overhead

---

Phase 4: Lane Unification (V68 - V80?) [IN PROGRESS]

Objective: Merge Lexical and Semantic lanes into a unified centroid-addressable manifold to improve efficiency and restore throughput.

Key Components:

- Centroid-based addressing
- Unified gating logic
- Reduced memory bandwidth (1 lane instead of 2)

Planned Experiments:

Version	Goal	Status	Target
V68.1	Lane unification + centroid gating	🔄 Planning	Restore throughput to >100k tok/s
V68.2	Optimize gating for recall	📅 Planned	90%+ rare recall @ 100k context
V68.3	Common recall at scale	📅 Planned	85%+ common recall @ 512k context
V68.4	Stress test at 512k	📅 Planned	Validate all metrics at target

Success Criteria (Phase):

- ☐ Throughput: >100k tok/s @ 512k context
- ☐ Rare Recall: >90% @ 512k
- ☐ Common Recall: >85% @ 512k

- [ ] Semantic Recall: >99% @ 512k
- [ ] Dictionary Recall: 100%
- [ ] Gate Density: <15%

Hypothesis:

- Unifying lanes reduces memory ops by 50% → throughput recovery
- Centroid gating allows "fuzzy" semantic matches to satisfy lexical queries → better slot utilization

Risks:

- Centroid approach might struggle with rare tokens (no clear prototype)
- Common recall might still fail at 512k (signal-to-noise ratio)

Next Phase Trigger: If V68 succeeds → Scale to 1B params. If V68 fails → Pivot to [TBD]

Phase 5: Scaling & Publication (V81+?) [FUTURE]

Objective: Scale validated architecture to production size and prepare for publication.

Planned Milestones:

- [ ] Scale to 1B parameters
- [ ] Scale to 3B parameters
- [ ] Comparative baselines (Memorizing Transformers, RMT)
- [ ] Multi-dataset validation (arXiv, Code, GSM8K)
- [ ] NeurIPS/ICLR submission
- [ ] Open-source release

Parking Lot 🅿️

Ideas for Future Exploration (not immediately critical):

- Multi-head gating (different gates for different attention heads)
- Hierarchical memory (short-term + long-term slots)
- Compression-based gating (only write if information gain is high)
- Dynamic slot allocation (adaptive memory size based on context length)

Update Protocol 📝

After each experiment, the assigned agent must:

Required Tool Calls:

1. `replace_file_content`: Update relevant Phase table (mark experiment status)
2. `replace_file_content`: Add learnings to "Learnings" subsection
3. `replace_file_content`: Check/uncheck success criteria
4. `replace_file_content`: Update "Current Status" at top
5. **If phase transition**: `replace_file_content`: Create new Phase section header

Example Update:

```
**Key Experiments**:
| Version | Goal | Status | Outcome |
|---|---|---|---|
| V68.1 | Lane unification | ✅ Complete | 112k tok/s, 91% rare recall |
```

Commit Message:

```
git commit -m "ROADMAP: Update after V68.1 - Lane unification complete"
```

When to pivot:

- If 3+ consecutive experiments fail → Call `notify_user` to discuss phase hypothesis
- If new insight invalidates strategy → Document pivot in journal, update roadmap phase