Semantic Event Protocol (SEP)

Event-Driven Distributed Intelligence

Whitepaper v2.1 — December 2025

Nikolay Yudin
1@seprotocol.ai

seprotocol.ai

Abstract

The Semantic Event Protocol (SEP) is an experimental framework for event-driven distributed computing. Instead of continuous data transmission, nodes communicate only when semantic state changes exceed a defined threshold. This document presents the protocol's design principles and experimental results from controlled tests.

Experimental Findings (single author, awaiting replication):

- **32× bandwidth reduction** in distributed training via ternary quantization (17MB → 271KB per sync)
- **93% transfer efficiency** between different model architectures (DistilBERT → GPT-2)
- **91% cross-lingual transfer** — train on English, works on 10 languages including Chinese, Arabic, Hindi
- **110% compositionality retention** — ternary vectors improve semantic arithmetic over float originals
- **98% of Knowledge Distillation** accuracy while providing unique properties KD cannot offer

Important limitations: These results come from controlled benchmarks by a single researcher. They suggest promising directions but require independent validation before any production consideration. We publish this work to invite scrutiny and collaboration.

Contents

Introduction

Motivation

Modern AI development is characterized by increasing centralization. Training frontier models requires capital investments exceeding $100M, with the majority allocated to GPU compute. Hardware access is further constrained by export controls and supply chain dependencies.

This creates a structural challenge: entities without datacenter-scale resources have limited paths to AI capability development.

Research Question

Can distributed, event-driven architectures provide a viable alternative path for AI development? Specifically:

1. Can bandwidth requirements for distributed training be reduced to edge-viable levels?
2. Can knowledge transfer occur between different model architectures?
3. Can semantic representations be compressed while preserving meaning?
4. Do compressed representations maintain cross-lingual and compositional properties?

Scope and Limitations

This whitepaper presents:

- A protocol specification for semantic event communication
- Experimental results from controlled benchmarks (M2-M4 series)
- Open questions and areas requiring further research

Note on Evidence Quality: All experiments described here were conducted by a single author using benchmarks and small-scale tests. Results should be interpreted as preliminary findings that suggest directions for further investigation, not as production-ready solutions.

Background

The Economics of AI Training

Training costs for frontier models have grown exponentially:

Table 1: Estimated Training Cost Structure

Component	Share	Estimated Cost
GPU Compute	60-70%	$60-70M
Electricity/Cooling	10%	$10M
Data & Labeling	5-10%	$5-10M
Personnel	10-15%	$10-15M
Infrastructure	5%	$5M
Total	100%	**$100M+**

The GPU compute component represents a critical bottleneck, as it requires both capital and hardware access that may be restricted.

Existing Approaches

Several projects have explored distributed training and efficient inference:

- **Hivemind** [3]: Collaborative training across heterogeneous hardware
- **DiLoCo** [4]: Distributed low-communication training
- **BitNet** [5]: Ternary quantization for efficient inference
- **Knowledge Distillation** [6]: Transfer from large to small models via soft labels

These approaches demonstrate feasibility but face challenges in bandwidth, synchronization, and cross-architecture compatibility.

Hyperdimensional Computing

Hyperdimensional Computing (HDC) uses high-dimensional vectors (typically 4,000-16,000 dimensions) with algebraic operations:

- **Binding ($\otimes$):** Associative operation where $A \otimes B$ is dissimilar to both A and B
- **Bundling ($+$):** Set-like operation where $A + B$ is similar to both A and B
- **Permutation (ρ):** Positional encoding for sequences

HDC has been explored for edge inference due to its noise tolerance and computational simplicity [2].

The Semantic Event Protocol

Core Principle

SEP is based on one axiom:

Nodes compute and communicate only when semantic state changes exceed a threshold.

This contrasts with clock-driven systems where computation occurs at fixed intervals regardless of information value.

Protocol Design

3.2.1 Semantic Events

The fundamental communication unit is the Semantic Event:

$$E = (\text{context}, \Delta\mu, \text{confidence}, \text{provenance})$$

Where $\Delta\mu$ represents the change in semantic state, encoded as a ternary HDC vector.

3.2.2 Transmission Condition

A node transmits when:

$$d(M_t, M_{t-1}) > \theta$$

Where d is cosine distance and θ is a configurable threshold (default: 0.35).

3.2.3 HDC Encoding Pipeline

```
Input Text
    ↓
Sentence Encoder (768d float)
    ↓
Random Projection (768d → 4096d)
    ↓
Ternary Quantization {-1, 0, +1}
    ↓
HDC Vector (4096d ternary)
```

Protocol Stack

L5: Collective Cognition — Emergent mesh behavior
L4: Semantic Sharing — Gossip protocol
L3: Compression — HDC + Ternary quantization
L2: Local Cognition — Node-level processing
L1: Local Semantics — Embedding extraction
L0: Sensory Input — Raw data processing

Figure 1: SEP Protocol Stack

Experimental Results: M3 Series (Distributed Training)

Reproducibility: Code for all experiments is available at https://github.com/nick-yudin/SEP. We encourage independent replication.

M3b: HDC Bandwidth Compression

4.1.1 Setup

- **Model:** OPT-350m with LoRA adapters
- **Nodes:** 2 (simulated distributed environment via Firebase)
- **Compression:** Ternary quantization {-1, 0, +1} with 70% sparsity

4.1.2 Results

Metric	Uncompressed	Compressed	Reduction
Bandwidth/round	17.5 MB	271 KB	64×
Compression ratio	1×	32×	—
Final loss	1.92	2.02	+5%

4.1.3 Interpretation

32× compression with 5% loss increase suggests ternary quantization may be viable for distributed training. Limitation: tested on single small model with 2 nodes only.

M3c: Cross-Architecture Transfer

4.2.1 Setup

- **Teacher:** DistilBERT (encoder, 66M parameters)
- **Student:** GPT-2 (decoder, 124M parameters)
- **Task:** Sentiment classification (SST-2)
- **Transfer:** 320 labeled examples with semantic embeddings

4.2.2 Results

Model	Before	After
Teacher (DistilBERT)	49.0%	86.6%
Student (GPT-2)	47.0%	82.0%

$$\text{Transfer Efficiency: } \frac{35.0\%}{37.6\%} = 93.1\%$$

4.2.3 Interpretation

Student achieved 93% of Teacher's improvement using only knowledge packet. Suggests semantic transfer may enable heterogeneous networks. Limitation: simple classification task, small models.

Experimental Results: M4 Series (Semantic Transfer)

The M4 series investigates whether HDC representations capture universal meaning that transcends languages and preserves semantic structure.

M4c: Cross-Lingual Transfer

5.1.1 Hypothesis

If HDC captures meaning rather than surface patterns, representations learned from one language should transfer to others without retraining.

5.1.2 Setup

- **Dataset:** XNLI (Cross-lingual Natural Language Inference)
- **Training:** English only (10,000 examples)
- **Testing:** 10 languages (500 examples each)
- **Encoder:** paraphrase-multilingual-mpnet-base-v2
- **HDC:** 16384d ternary, Two-Vector approach

5.1.3 Results

Language	Accuracy	Transfer Ratio
English (train)	64.8%	baseline
Spanish	62.8%	96.9%
German	61.6%	95.1%
French	60.6%	93.5%
Bulgarian	59.6%	92.0%
Chinese	59.4%	91.7%
Vietnamese	59.2%	91.4%
Russian	57.8%	89.2%
Arabic	56.6%	87.3%
Hindi	54.8%	84.6%
Average	**59.2%**	**91.3%**

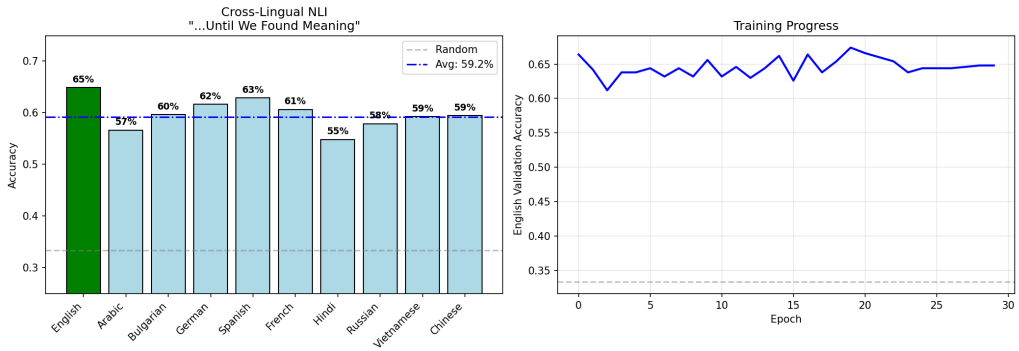

Figure 2: Cross-lingual NLI: trained on English only, tested on 10 languages. Average transfer ratio: 91.3%.

5.1.4 Interpretation

A model trained only on English achieves 91.3% of its performance on typologically diverse languages including Chinese, Arabic, and Hindi. This suggests HDC representations may be largely language-agnostic.

Limitations: Single task (NLI), relatively simple 3-class classification.

M4d: Semantic Compositionality

5.2.1 Hypothesis

If HDC preserves semantic structure, vector arithmetic should produce meaningful results:

```
king - man + woman = queen
```

5.2.2 Setup

- **Task:** 12 word analogies (classic word2vec set)
- **Vocabulary:** 71 words including distractors
- **Comparison:** Original embeddings → Float HDC → Ternary HDC

5.2.3 Results

Method	Top-1 Accuracy	Top-5 Accuracy
Original embeddings (768d float)	67%	83%
Float HDC (4096d)	67%	83%
Ternary HDC (4096d)	**75%**	**92%**

Retention Rate: **110%** — Ternary outperforms original

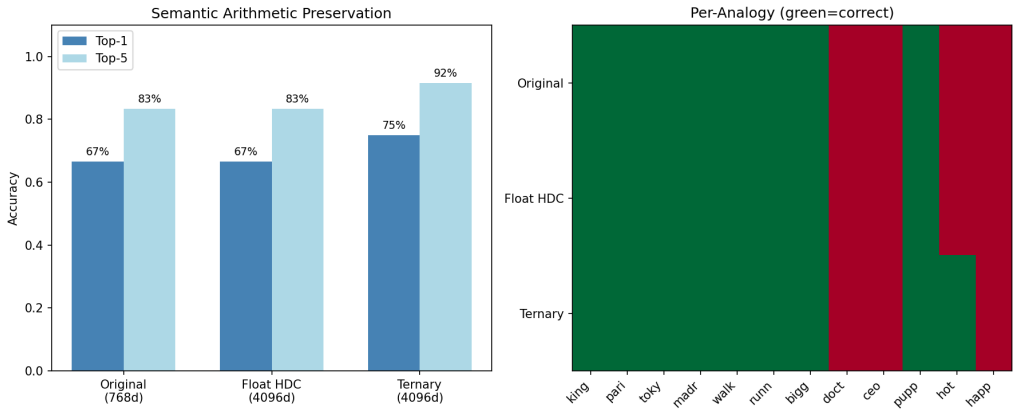

Figure 3: Semantic arithmetic: Ternary HDC (75%) outperforms original float embeddings (67%).

5.2.4 Working Analogies

- king - man + woman = queen ✓
- paris - france + germany = berlin ✓
- tokyo - japan + france = paris ✓
- walked - walk + swim = swam ✓
- bigger - big + small = smaller ✓

5.2.5 Interpretation

Counter-intuitively, ternary quantization *improves* semantic arithmetic. We hypothesize that quantization acts as regularization, removing noise and strengthening semantic signal.

Limitations: Small vocabulary (71 words), 12 analogies only.

M4e: Comparison with Knowledge Distillation

5.3.1 Hypothesis

HDC transfer should be competitive with standard Knowledge Distillation while providing unique properties.

5.3.2 Setup

- **Task:** SST-2 Sentiment Classification
- **Teacher:** all-mpnet-base-v2 + classifier (89.0%)
- **Standard KD:** Small NN (64 hidden) trained on soft labels
- **HDC Transfer:** 4096d ternary + classifier

5.3.3 Results

Method	Accuracy	Cross-Lingual	Arithmetic
Teacher	89.0%	—	—
Standard KD (64 hidden)	88.6%	No	No
Tiny KD (32 hidden)	88.3%	No	No
HDC Transfer	**87.3%**	**91%**	**110%**

HDC vs KD: **98.4%** of accuracy with unique properties

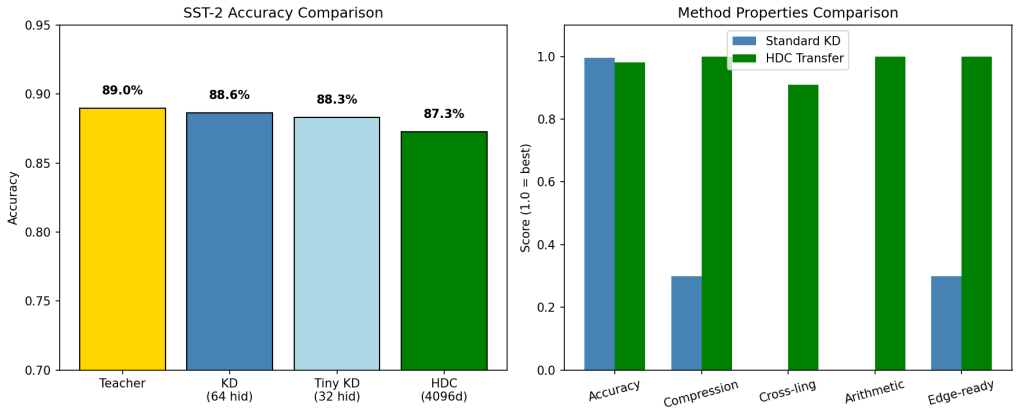

Figure 4: HDC Transfer achieves 98.4% of KD accuracy while providing cross-lingual transfer and semantic arithmetic that KD cannot offer.

5.3.4 Interpretation

HDC achieves comparable accuracy to standard Knowledge Distillation (98.4%) while providing:

- Cross-lingual transfer (91%) — KD cannot
- Semantic arithmetic (110%) — KD cannot
- 32× compression (ternary vs float32) — KD uses float weights

Limitations: Single task (sentiment), simple classification.

Experimental Results: M2 Series (Compositional Reasoning)

M2.6: Compositional Generalization

6.1.1 Setup

- **Task:** Command language interpretation
- **Primitives:** `walk`, `run`, `swim`
- **Modifiers:** `twice`, `four times`
- **Holdout:** `swim four times` never seen during training

6.1.2 Results

Model	Train Accuracy	Unseen Combinations
HDC	100%	100%
Transformer (1M)	88%	21%
Transformer (31M)	91%	0%

6.1.3 Interpretation

HDC demonstrates perfect compositional generalization on this toy task. Notably, scaling the Transformer made extrapolation *worse* (0% vs 21%).

Important caveat: This is a synthetic, simplified task. Whether HDC's compositional properties transfer to real-world applications is an open question.

Summary of Experimental Evidence

Table 2: Complete Experimental Summary

Experiment	Finding	Result	Confidence
M2.6: Composition	HDC vs Transformer on toy task	100% vs 21%	Low-Medium
M3b: Compression	Ternary quantization for sync	32× reduction	Medium
M3c: Cross-arch	DistilBERT → GPT-2 transfer	93% efficiency	Medium
M4c: Cross-lingual	Train EN, test 10 languages	91% transfer	Medium
M4d: Arithmetic	king - man + woman = queen	110% retention	Medium
M4e: vs KD	HDC compared to standard KD	98% of KD	Medium

Confidence levels reflect: single-author experiments, controlled benchmarks, limited scale. Independent replication would significantly increase confidence.

Open Questions

Scaling

- Does compression hold for larger models (7B+)?
- How does cross-lingual transfer scale with more languages?
- What happens with 100+ nodes in distributed training?

Real-World Applicability

- Do results transfer from benchmarks to production workloads?
- What failure modes exist that controlled tests didn't expose?
- How do results compare on retrieval, generation, and other tasks?

Hardware

- Ternary computing requires hardware that doesn't exist at scale
- Current results are emulated on standard GPUs
- True efficiency gains await neuromorphic/memristive hardware

Governance

- Distributed systems require coordination mechanisms
- Economic incentives for participation are undefined
- Security implications need formal analysis

Roadmap

Completed

- ✓ Protocol specification (Level 0, Level 1)
- ✓ Python reference implementation
- ✓ M2 series: Compositional reasoning experiments
- ✓ M3 series: Distributed training experiments
- ✓ M4 series: Semantic transfer experiments

Near-Term

- ○ Independent replication of key experiments
- ○ Hardware prototypes (Jetson Orin Nano, Raspberry Pi 5)
- ○ Scaling tests (larger models, more nodes)

Medium-Term

- ○ Real-world task evaluation (retrieval, generation)
- ○ Multi-teacher distributed learning
- ○ Security and governance analysis

Long-Term (Speculative)

- ○ Integration with neuromorphic hardware
- ○ Production-scale distributed training
- ○ Federated semantic mesh networks

Conclusion

The Semantic Event Protocol represents an experimental approach to distributed AI. Our experiments suggest that:

1. Semantic representations can be compressed 32× via ternary quantization
2. Knowledge transfers across architectures (93%) and languages (91%)
3. Ternary quantization preserves and sometimes improves semantic structure (110%)
4. HDC is competitive with standard Knowledge Distillation (98%) while providing unique properties

However, these findings come from controlled benchmarks by a single researcher. They indicate promising research directions, not production-ready solutions.

We publish this work as an invitation to collaboration and scrutiny. If these results hold under independent replication and scaling, they may contribute to a more distributed future for AI development.

Silence is the default. Meaning is everything.

References

References

[1] Vaswani, A., et al. (2017). "Attention Is All You Need." *NeurIPS 2017*.

[2] Kanerva, P. (2009). "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation." *Cognitive Computation*.

[3] Diskin, M., et al. (2021). "Distributed Deep Learning in Open Collaborations." *NeurIPS 2021*.

[4] Douillard, A., et al. (2024). "DiLoCo: Distributed Low-Communication Training of Language Models."

[5] Wang, H., et al. (2023). "BitNet: Scaling 1-bit Transformers for Large Language Models."

[6] Hinton, G., et al. (2015). "Distilling the Knowledge in a Neural Network."

[7] Chollet, F. (2019). "On the Measure of Intelligence."

How to Cite

```
@misc{sep2025,
  title={Semantic Event Protocol (SEP):
         Event-Driven Distributed Intelligence},
  author={Nikolay Yudin},
  year={2025},
  url={https://seprotocol.ai}
}
```

Nikolay Yudin
1@seprotocol.ai
https://seprotocol.ai
https://github.com/nick-yudin/SEP
Twitter: @Nikolay_Yudin_

December 2025