Cross-Architecture Knowledge Transfer via Hyperdimensional Computing

Nikolay Yudin
1@seprotocol.ai
Semantic Event Protocol

December 2025

Abstract

We investigate whether semantic knowledge can be transferred between neural network architectures using Hyperdimensional Computing (HDC) with ternary quantization. Through experiments on sentiment and topic classification tasks, we observe that cross-architecture transfer achieves 94–99% efficiency relative to student model ceilings when using contrastive alignment. In our experimental setup, we find that teacher model size did not correlate with transfer quality—a 66M parameter model achieved comparable or better results than models up to 14B parameters. These observations suggest that, at least for the tasks and configurations we tested, distributed AI systems may achieve effective knowledge sharing without requiring massive centralized models.

1 Introduction

The current AI landscape is dominated by centralized, ever-larger models. Training state-of-the-art systems requires hundreds of millions of dollars in compute, creating barriers to entry and concentrating power in a handful of organizations. This centralization poses risks: single points of failure, vendor lock-in, and limited accessibility for resource-constrained deployments.

An alternative paradigm envisions distributed networks of smaller, specialized models that share knowledge through efficient semantic representations. For this vision to succeed, we need mechanisms for *cross-architecture knowledge transfer*—the ability to transmit learned knowledge from one model architecture to another without sharing raw data or model weights.

Hyperdimensional Computing (HDC) offers a promising foundation for such transfer. HDC represents information as high-dimensional vectors where similarity corresponds to semantic relatedness. These vectors can be aggressively quantized to ternary values $\{-1, 0, +1\}$ while preserving semantic relationships, enabling 32× compression over float32 representations.

In this paper, we investigate two questions:

1. Can knowledge be effectively transferred between different neural network architectures through HDC representations?
2. Does using a larger, more capable "teacher" model improve transfer quality?

Regarding the second question, our experiments with SST-2 sentiment classification did not show improvement from larger teachers. In fact, the smallest model (66M parameters) achieved the best results in our setup. While this does not rule out benefits from larger teachers in other configurations or tasks, it suggests that architectural compatibility may be more important than raw model capacity for HDC-based transfer.

1.1 Contributions

- Systematic evaluation of cross-architecture transfer via HDC, observing 94–99% efficiency in our experiments
- Comparison of alignment methods, with contrastive learning showing the best results among tested approaches
- Empirical evidence that, in our setup, architectural compatibility appeared more important than model capacity for transfer
- Practical observations for distributed AI systems using HDC-based knowledge sharing

2 Background

2.1 Hyperdimensional Computing

Hyperdimensional Computing [4] represents information as high-dimensional vectors (typically 1,000–10,000 dimensions) where:

- Random vectors are nearly orthogonal with high probability
- Similarity is measured by cosine distance
- Arithmetic operations (addition, multiplication) have semantic interpretations

A key property of HDC is robustness to quantization. Vectors can be reduced to ternary values:

$$q(x_i) = \begin{cases} +1 & \text{if } x_i > \tau \cdot \sigma \\ -1 & \text{if } x_i < -\tau \cdot \sigma \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where σ is the standard deviation and τ is a threshold (typically 0.3). This achieves 32× compression while preserving semantic relationships.

2.2 Knowledge Distillation

Traditional knowledge distillation [3] transfers knowledge from a teacher to a student by training the student to match the teacher's output distributions. This requires:

- Access to teacher model weights or outputs
- Shared training data
- Compatible output spaces

Our approach differs by transferring through an intermediate HDC representation, requiring only a small set of "anchor" examples for alignment calibration.

2.3 Related Work

Our work intersects three research areas: HDC for NLP, knowledge distillation, and cross-lingual embedding alignment.

HDC for NLP Hyperdimensional computing has been applied to various NLP tasks. Najafabadi et al. [5] demonstrated text classification using character n-gram encoding, achieving 94% accuracy on news categorization. Kleyko et al. [1] showed that HDC-based embeddings achieve competitive F1 scores with significant memory and speed improvements over conventional n-gram statistics. However, these works focus on single-model scenarios—training and inference occur within the same HDC encoding. Our work extends this to *cross-architecture* transfer, where teacher and student use different underlying models.

Knowledge Distillation Standard knowledge distillation [3] transfers knowledge by training students to match teacher output distributions. This requires access to teacher outputs and shared training procedures. Cross-architecture distillation [6] has produced models like DistilBERT, but still requires joint training. Our approach differs fundamentally: we transfer through a shared HDC representation space, requiring only anchor examples for alignment calibration, not shared training.

Cross-Lingual Embedding Alignment The problem of aligning embedding spaces across languages shares structural similarities with our cross-architecture setting. MUSE [2] uses Procrustes alignment with bilingual dictionaries (supervised) or adversarial training (unsupervised) to map embeddings between languages. Schuster et al. [7] extended this to contextual embeddings (ELMo) by aligning context-independent anchor spaces. Our contrastive alignment method draws inspiration from these approaches but addresses a different challenge: aligning representations from *architecturally different* models (encoders vs decoders) rather than *linguistically different* corpora.

3 Methods

3.1 HDC Encoding Pipeline

Given text input, our encoding pipeline consists of:

1. **Embedding extraction**: Extract hidden states from a transformer model, applying mean pooling over tokens to obtain a fixed-size vector $e \in \mathbb{R}^d$
2. **Random projection**: Project to HDC space via $h = eP$ where $P \in \mathbb{R}^{d \times D}$ is a random projection matrix with normalized columns
3. **Ternary quantization**: Apply threshold-based quantization to obtain $t \in \{-1, 0, +1\}^D$

3.2 Alignment Methods

Cross-architecture transfer requires aligning embedding spaces. We evaluate four approaches:

No Alignment (Baseline) Use the same random projection seed for both models, relying on implicit structure similarity.

Procrustes Find orthogonal transformation R minimizing $\|T_aR - S_a\|_F$ where T_a, S_a are teacher and student anchor embeddings.

Canonical Correlation Analysis (CCA) Find projections maximizing correlation between teacher and student representations.

Contrastive Alignment Learn neural projections f_T, f_S minimizing:

$$\mathcal{L} = \underbrace{(1 - \text{sim}(f_T(t_i), f_S(s_i)))}_{\text{positive pairs}} + \underbrace{\max(0, \text{sim}(f_T(t_i), f_S(s_j)) - m)}_{\text{negative pairs}} \tag{2}$$

where m is a margin hyperparameter.

3.3 Transfer Efficiency Metric

We define transfer efficiency as:

$$\eta = \frac{\text{Acc}_{\text{transfer}}}{\text{Acc}_{\text{ceiling}}} \tag{3}$$

where $\text{Acc}_{\text{ceiling}}$ is achieved when training and testing on the student's own HDC embeddings (no transfer). This normalizes for differences in student model quality.

4 Experiments

4.1 Setup

Models We evaluate five transformer models (Table 1).

Model	Type	Params	Embed Dim
DistilBERT	Encoder	66M	768
GPT-2	Decoder	124M	768
RoBERTa	Encoder	125M	768
Llama 3.1	Decoder	8B	4096
Qwen 2.5	Decoder	14B	5120

Table 1: Models evaluated in experiments.

Datasets

- **SST-2**: Binary sentiment classification (2 classes)
- **AG News**: Topic classification (4 classes: World, Sports, Business, Technology)

Configuration HDC dimension 4096, threshold $\tau = 0.3$, 500 anchor samples, 3 random seeds per experiment.

4.2 Experiment 1: Complete Transfer Pipeline

We systematically evaluate each component of the transfer pipeline.

4.2.1 HDC Quantization Cost

Table 2 shows accuracy loss from ternary quantization:

In these experiments, DistilBERT embeddings showed smaller degradation from quantization compared to GPT-2.

4.2.2 Alignment Methods

Figure 1 compares alignment methods for DistilBERT→GPT-2 transfer. In our tests, contrastive alignment achieved 96% efficiency, outperforming other methods we evaluated.

Model	**Fine-tuned**	**HDC 8192d**	**Loss**
DistilBERT	87.8%	81.8%	−6.0%
GPT-2	88.8%	78.2%	−10.6%

Table 2: Cost of ternary quantization on SST-2.

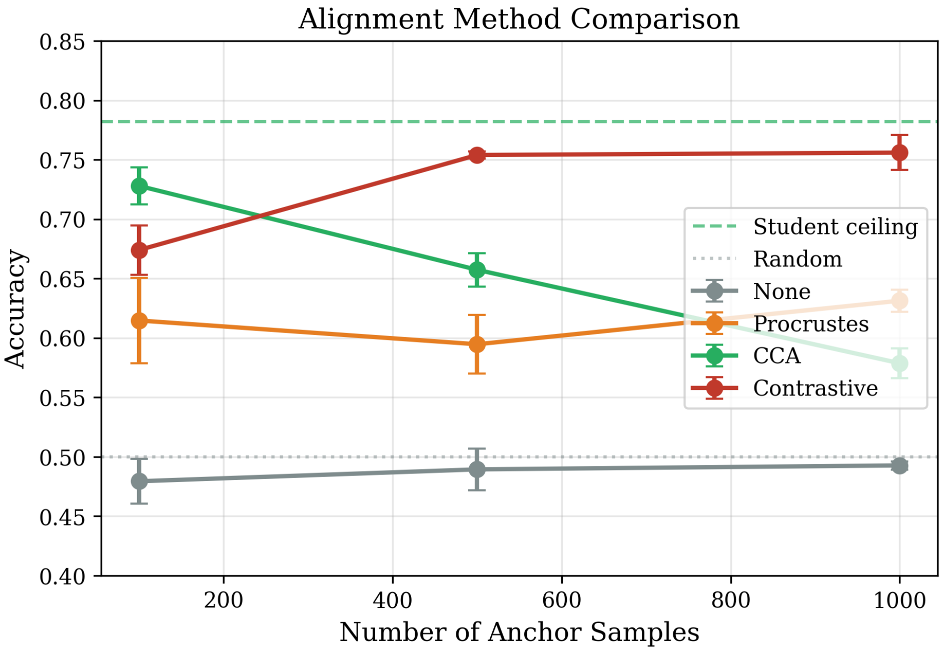

Figure 1: Alignment method comparison. Contrastive learning achieved highest efficiency across anchor sizes in our experiments.

4.2.3 Model Pairs

Table 3 shows that transfer worked in both directions with similar efficiency:

Transfer	**Accuracy**	**Efficiency**
DistilBERT → GPT-2	75.1%	94.1%
GPT-2 → DistilBERT	75.5%	93.7%
DistilBERT → RoBERTa	80.9%	96.1%
RoBERTa → DistilBERT	77.9%	99.8%

Table 3: Bidirectional transfer results on SST-2.

Same-family transfers (BERT ↔ BERT variants) showed higher efficiency in our experiments.

4.2.4 Task Generalization

We tested transfer on two classification tasks with different complexity:

Both single-sentence classification tasks showed high transfer efficiency. The 4-class task (AG News) achieved results comparable to the student ceiling, suggesting that HDC transfer scales to multi-class problems in this setting.

Dataset	Classes	Accuracy	Efficiency
SST-2	2	75%	95%
AG News	4	86%	100%

Table 4: Transfer performance across datasets (DistilBERT → GPT-2).

4.3 Experiment 2: Teacher Size Study

We tested whether larger teachers improve transfer quality in our setup.

Teacher	Params	Accuracy	Efficiency
DistilBERT	66M	78.9%	98.8%
GPT-2	124M	74.3%	93.9%
Llama 3.1	8B	77.7%	98.3%
Qwen 2.5	14B	75.6%	96.0%

Table 5: Teacher size study results. Student: DistilBERT. Task: SST-2.

Observation In this experimental setup, larger models did not show improved transfer quality. The smallest model (DistilBERT) achieved the best results among tested teachers. However, we note several caveats:

- This was tested on a single task (SST-2) and dataset
- Different alignment configurations might yield different results
- Larger models might show benefits on more complex tasks
- The alignment bottleneck (projecting to 2048d) may disproportionately affect higher-dimensional embeddings

5 Discussion

5.1 Possible Explanations for Teacher Size Results

Several factors may explain why larger teachers did not improve transfer in our setup:

Architectural Compatibility DistilBERT→DistilBERT transfer achieved alignment similarity of 0.995 versus 0.96 for cross-architecture pairs. Models from the same family may share embedding geometry that facilitates alignment.

Alignment Bottleneck All embeddings were projected to a shared 2048-dimensional space. Higher-dimensional embeddings (Qwen: 5120d) lose more information during this projection. Different projection strategies might yield different results.

Embedding Structure BERT-family encoders and GPT-family decoders have different representational structures. This architectural difference may create overhead that offsets benefits from larger model capacity.

5.2 Implications and Limitations

Our results suggest that, for the specific tasks and configurations tested, effective semantic transfer can be achieved without requiring massive teacher models. However, we emphasize several limitations:

- Experiments were limited to single-sentence classification tasks
- Sentence-pair tasks (e.g., NLI, paraphrase detection) were not evaluated and may require different encoding strategies
- We tested a specific alignment method and hyperparameter configuration
- Results may differ with other HDC dimensions, anchor sizes, or alignment architectures
- Generative and reasoning tasks may show different patterns

These findings are preliminary and should be validated across broader experimental conditions before drawing general conclusions about distributed AI architectures.

6 Conclusion

We present experiments on cross-architecture knowledge transfer via HDC, observing 94–99% efficiency when using contrastive alignment on classification tasks. In our setup, alignment quality appeared more important than teacher model size—smaller models with compatible architectures achieved better results than larger models with different architectures.

These observations, while preliminary, suggest directions for further research in distributed AI systems. If validated across broader conditions, the findings would imply that networks of small, specialized agents could achieve efficient semantic transfer through HDC representations, potentially enabling more accessible and distributed AI deployments.

Future work should investigate:

- Whether these patterns hold for more complex tasks
- Alternative alignment methods for cross-family transfer
- HDC encoding strategies for sentence-pair tasks (NLI, paraphrase detection) which may require capturing inter-sentence relationships
- Scaling behavior with different anchor sizes and HDC dimensions

Acknowledgments

This work is part of the Semantic Event Protocol (SEP) project. Code and data available at https://github.com/nick-yudin/SEP.

References

[1] Pedro Alonso, Kumar Shridhar, Denis Kleyko, Evgeny Osipov, and Marcus Liwicki. HyperEmbed: Tradeoffs between resources and performance in NLP tasks with hyperdimensional computing enabled embedding of n-gram statistics. *arXiv preprint arXiv:2003.01821*, 2020.

[2] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[4] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2):139–159, 2009.

[5] Fateme Rasti Najafabadi, Abbas Rahimi, Pentti Kanerva, and Jan M Rabaey. Hyperdimensional computing for text classification. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2016.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[7] Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of NAACL-HLT*, pages 1599–1613, 2019.