R-JEPA: Predictive Relation-Space Learning for Retrieval and Within-Label Structure

Anonymous ACL submission

Abstract

We study relation representation learning from prediction rather than direct label separation. We propose **R-JEPA**, a joint-embedding predictive framework that learns to predict a target relation representation from relational context. We evaluate it on **Re-DocRED**, a revised document-level relation extraction benchmark that better supports retrieval- and structure-based analysis under reduced annotation incompleteness. Predictive training does not induce a single best geometry. Full R-JEPA gives the best relation retrieval, while prediction-only training yields much stronger within-label structure and clearer interpretable subgroups. Context ablations further show that, in our frozen-span setup, these differences are not driven mainly by better document-context use. Instead, they come mostly from how the objective reshapes relation space on top of fixed base representations. These results suggest that predictive relation learning is a useful framework for document-level relation modeling and that retrieval quality and label-aligned internal structure should be treated as distinct properties of relation space.

1 Introduction

Relations are a basic unit of reasoning and have long been studied in analogy, relation extraction, relational similarity, and distributional inference (Gentner, 1983; Lin and Pantel, 2001; Turney, 2006; Baldini Soares et al., 2019; Rossiello et al., 2019; Ushio et al., 2021, 2025). Systems that read text or structured knowledge must identify relations between entities, compare relation instances, retrieve structurally similar examples, and support analogy-style generalization. These abilities matter in document-level relation extraction, relational search, exemplar-based comparison, and analogy-style reasoning (Yao et al., 2019; Han et al., 2018; Baldini Soares et al., 2019; Rossiello et al., 2019; Ushio et al., 2025).

Most existing approaches learn relation representations through direct label prediction or discriminative separation of positive and negative instances (Baldini Soares et al., 2019; Zhou et al., 2021; Wang et al., 2023). In document-level relation extraction, this has produced strong models based on multi-sentence reasoning, graph-structured inference, mention interaction, and localized evidence aggregation (Christopoulou et al., 2019; Zeng et al., 2020; Xu et al., 2021; Zhou et al., 2021; Zhang et al., 2021). These objectives work well for fixed-label classification, but they do not necessarily yield a relation space organized for retrieval, latent structure, or broader relational generalization.

We study a predictive alternative. Inspired by joint-embedding predictive learning, we predict a target relation representation from context in latent space rather than directly predicting a discrete label (van den Oord et al., 2018; Assran et al., 2023; Huang et al., 2026). We call this framework **R-JEPA**. Our goal is not to recover a universal ontology of relations, but to examine the geometry induced by predictive learning.

We evaluate on **Re-DocRED** (Tan et al., 2022). Re-DocRED is suitable for this setting because relation identity is expressed in document context, it preserves the multi-sentence reasoning setting of DocRED, and its revised annotations reduce incompleteness in the original dataset (Yao et al., 2019; Tan et al., 2022). This is important because we evaluate not only retrieval, but also internal geometry, within-label subgroup structure, and analogy-style neighborhoods.

Recent document-level relation extraction work has increasingly emphasized document-level reasoning, relation interactions, and retrieval- or filtering-oriented pipelines rather than only local pairwise classification (Christopoulou et al., 2019; Zeng et al., 2020; Xu et al., 2021; Zhou et al., 2021; Zhang et al., 2021, 2025a,b). Work on open or unsupervised relation extraction has likewise high-

lighted relation-space organization beyond fixed label prediction, including clustering quality, relation discovery, and uncertainty-aware interpretation (Zhao et al., 2021, 2023; Wang et al., 2025; Jamal et al., 2025). Re-DocRED therefore provides a useful testbed for asking not only whether a model predicts the correct label, but also how its relation space is organized.

Our central claim is simple: predictive training does not induce a single best relation geometry. Instead, it improves different properties of relation space in different ways. Full R-JEPA is strongest for retrieval, whereas prediction-only training is strongest for within-label structure.

This paper makes three contributions. First, we formulate document-level relation learning as context-to-target prediction in embedding space rather than direct label separation. Second, on Re-DocRED, we show a split between retrieval-oriented and structure-oriented geometries: full R-JEPA performs best for retrieval, whereas prediction-only training induces much stronger within-label structure. Third, we show through subgroup and analogy-style analyses that predictive relation spaces support interpretable organization beyond exact label matching.

2 Related Work

2.1 Relation Learning and Relation Extraction

A large body of work studies relation learning through supervised relation extraction, pair classification, and knowledge-guided prediction. In document-level relation extraction, prior work has emphasized multi-sentence reasoning, graph-based propagation, mention interaction, and context aggregation over long documents (Yao et al., 2019; Christopoulou et al., 2019; Zeng et al., 2020; Xu et al., 2021; Zhou et al., 2021; Zhang et al., 2021; Delaunay et al., 2023). More recent work has moved beyond local entity-pair classification to model global relation interactions, relation correlations, and entity-pair-guided retrieval or filtering mechanisms (Zhang et al., 2025a,b; Han et al., 2024). These methods are effective for fixed-label classification, but they do not necessarily encourage a relation space organized for retrieval or broader relational structure.

2.2 Relation Representation and Analogical Structure

A related line of work studies relation representations through analogy, retrieval, and structure-sensitive generalization. Classical work on relational similarity and distributional inference argued that relations can be compared independently of specific lexical items using distributional or structural regularities (Lin and Pantel, 2001; Turney, 2006; Gentner, 1983; Gentner and Markman, 1997). More recent work has shown that relation representations can be learned from textual mentions and evaluated through relational similarity, exemplar retrieval, few-shot comparison, or analogy-based generalization (Han et al., 2018; Baldini Soares et al., 2019; Rossiello et al., 2019; Ushio et al., 2021, 2025). These works show that relation representations can encode useful regularities, but they usually do not ask how the training objective shapes relation-space geometry.

2.3 Open and Unsupervised Relation Extraction

Another related line of work studies relation-space organization beyond fixed label prediction through clustering-based open relation extraction and unsupervised relation discovery. Prior work has emphasized that useful relation representations should support not only classification, but also relation discovery, cluster alignment, and interpretable subgroup formation (Zhao et al., 2021, 2023; Wang et al., 2023, 2025; Jamal et al., 2025). Our setting is supervised document-level relation learning rather than open relation discovery, but these studies motivate our focus on within-label structure, subgroup emergence, and relation-space geometry beyond top-line classification accuracy.

2.4 Predictive Representation Learning and JEPAs

Joint Embedding Predictive Architectures learn by predicting target representations from context in latent space rather than reconstructing raw observations (Assran et al., 2023). Related predictive representation learning has also been explored through latent prediction objectives such as contrastive predictive coding (van den Oord et al., 2018). More recent work has extended JEPA-style objectives beyond vision, including language modeling and retrieval-oriented representation learning (Huang et al., 2026; Choudhury et al., 2025). We adapt

a JEPA-style predictive framework to document-level relation representation learning and use it to study how predictive training organizes relation space differently from discriminative objectives.

2.5 Predictive and Discriminative Objectives

Contrastive and metric-learning objectives directly encourage separation between positives and negatives (van den Oord et al., 2018; Chen et al., 2020; Wang et al., 2023). Predictive objectives instead encourage context-sensitive regularities that help infer a held-out target representation (Assran et al., 2023; Huang et al., 2026). Our experiments ask what kinds of relation geometry these objectives induce.

3 Problem Formulation

We consider relation instances of the form

$$x_{ij} = (e_i, e_j, c_{ij}),$$

where e_i and e_j are two entities or mentions, and c_{ij} is the context in which the relation is expressed. In Re-DocRED, c_{ij} is document-level context. Our goal is to learn a relation representation

$$r_{ij} \in \mathbb{R}^d$$

that captures the relation between e_i and e_j.

Rather than predicting a discrete label directly, we formulate relation learning as context-to-target prediction in embedding space. Given a set of contextual relation instances

$$\mathcal{C} = \{x_{c_1}, x_{c_2}, \ldots, x_{c_k}\},$$

the model predicts the latent representation of a held-out target relation instance x_t.

4 R-JEPA

4.1 Overview

R-JEPA instantiates the predictive formulation above with three components. The encoder E maps a relation instance to a local pair representation, the relation head R projects that representation into relation space, and the predictor P uses sampled context relations to predict a held-out target relation embedding. Unlike standard relation classification, the model is trained to predict a target embedding rather than a discrete label. Training follows the JEPA pattern: an online branch makes the prediction, while a target branch provides a slowly updated stop-gradient target representation (Assran et al., 2023).

Figure 1 summarizes the pipeline. A relation instance is first encoded into relation space, after which sampled context relations are used to predict a held-out target relation in latent space. This predictive objective shapes the learned relation space, which we later analyze through retrieval, within-label subgroup structure, and cross-label analogy-style neighborhoods.

4.2 Pair Encoder and Relation Head

Each relation instance x_{ij} is encoded as

$$z_{ij} = E(x_{ij}).$$

In our document-level setting, the encoder consumes entity spans and their document context under a frozen contextual backbone. The pair-level representation is then mapped into relation space:

$$r_{ij} = R(z_{ij}).$$

The role of R is to emphasize relation-level structure rather than raw entity identity. If two instances share similar relational structure, their embeddings should be nearby or at least predictively compatible in the learned space.

4.3 Relational Context Construction

For each training episode, we construct a context set

$$\mathcal{C} = \{x_{c_1}, \ldots, x_{c_k}\}$$

and a held-out target relation x_t. In the current system, context is defined over sampled relation instances rather than explicit fine-grained sublabels. This lets us ask whether predictive training can shape relation space even when subgroup structure is not directly supervised.

4.4 JEPA-style Predictive Objective

Given context relation embeddings

$$\{r_{c_1}, \ldots, r_{c_k}\},$$

the predictor produces a target estimate:

$$\hat{r}_t = P(r_{c_1}, \ldots, r_{c_k}).$$

The target representation is computed by a stop-gradient target branch:

$$r_t = \mathrm{sg}(R_{\text{tar}}(E_{\text{tar}}(x_t))).$$

We minimize the prediction loss

$$\mathcal{L}_{\text{pred}} = \|\hat{r}_t - r_t\|_2^2.$$

This objective trains the model to infer a target relation representation from context rather than output a discrete relation label.

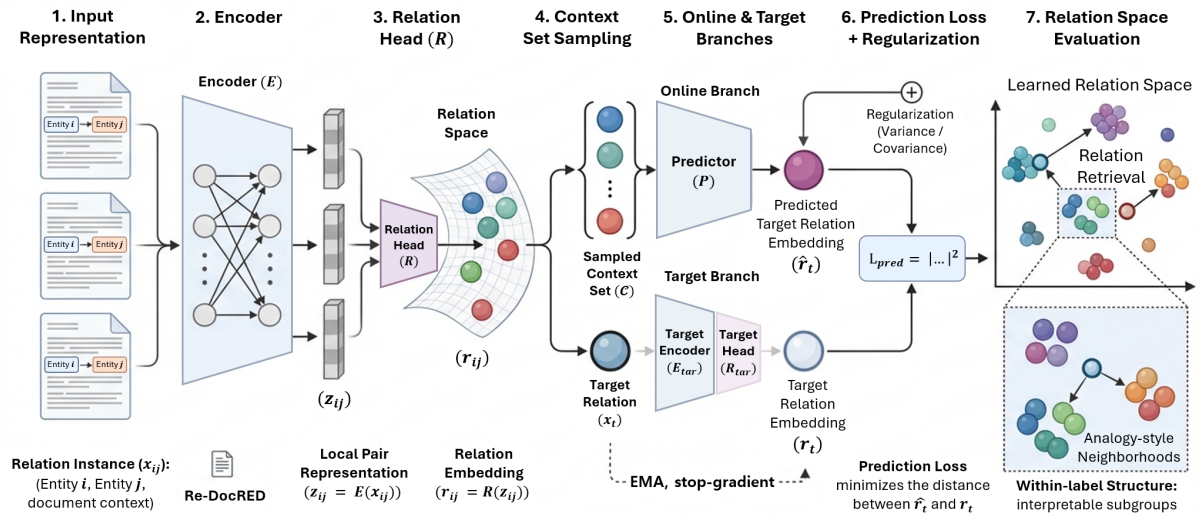

Figure 1: **Overview of R-JEPA.** Given a relation instance x_{ij}, the encoder E produces a local pair representation z_{ij}, which the relation head R maps into a relation embedding r_{ij}. During training, the predictor P uses a sampled context set $\mathcal{C}$ of relation embeddings to predict a held-out target relation embedding from a stop-gradient EMA target branch. The learned relation space is then evaluated for relation retrieval, within-label structure, and analogy-style neighborhoods.

4.5 Target Branch and EMA Update

The target branch is updated by exponential moving average:

$$\theta_{\text{tar}} \leftarrow m\theta_{\text{tar}} + (1-m)\theta_{\text{online}},$$

where m is a momentum coefficient. This asymmetry stabilizes learning: the target branch changes slowly, while the online branch learns to predict against that slower target.

4.6 Optional Regularization

We optionally add regularization terms to stabilize the learned geometry, following variance- and covariance-based anti-collapse regularization in self-supervised representation learning (Bardes et al., 2022):

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}} + \lambda_{\text{cov}}\mathcal{L}_{\text{cov}}.$$

The variance term prevents dimension collapse, while the covariance term discourages redundant dimensions. These regularizers are auxiliary; the core learning signal remains context-to-target prediction.

4.7 Relation Retrieval and Analogy-style Neighborhoods

Once the model has learned a relation space, retrieval is straightforward. For a query instance x_q, we compute

$$r_q = R(E(x_q)).$$

We then rank candidate relations by cosine similarity:

$$\text{score}(q, i) = \cos(r_q, r_i).$$

This yields **relation retrieval**. The same space can also support **analogy-style neighborhoods**. In that case, a retrieved instance need not share the exact label of the query. It may instead instantiate a structurally similar relation in a nearby region of relation space. In this paper, such neighborhoods are treated as a downstream property of the learned space rather than the training objective.

5 Experiments

5.1 Dataset

We use **Re-DocRED** as our only benchmark (Tan et al., 2022). Our goal is not broad benchmark coverage, but a controlled study of predictive document-level relation representation learning in a setting where retrieval, subgroup structure, and analogy-style neighborhoods can be analyzed on revised document-level annotations.

5.2 Baselines

We compare R-JEPA against an encoder-only baseline, a contrastive baseline, a contrastive baseline with regularization, and a prediction-only R-JEPA variant without auxiliary regularization. This comparison isolates the effect of predictive training from the effect of geometric regularization.

Model	MRR	R@1	R@5	ARI	Probe
Encoder-only	0.6060±0.0001	0.4101±0.0011	0.8888±0.0033	0.023±0.001	0.465±0.002
Contrastive + Reg.	0.6180±0.0026	0.4217±0.0007	0.9008±0.0044	0.032±0.000	0.517±0.006
R-JEPA (pred-only)	0.6037±0.0059	0.4031±0.0073	0.8901±0.0024	**0.084±0.008**	**0.607±0.005**
R-JEPA (full)	**0.6216±0.0012**	**0.4237±0.0010**	**0.9059±0.0034**	0.034±0.001	0.530±0.005

Table 1: Main results on Re-DocRED over three random seeds. Full R-JEPA gives the best retrieval, whereas prediction-only training gives the strongest internal structure.

5.3 Evaluation Protocol

We evaluate several properties of the learned relation space. Retrieval quality is measured with MRR, Recall@1, Recall@5, and Recall@10. Probe accuracy measures how much gold-label information is linearly decodable from frozen relation embeddings. ARI measures how closely induced clusters align with gold labels. For within-label substructure, we cluster instances only within the same gold relation label and compute aggregate statistics such as silhouette values and the number of labels that satisfy our current substructure criterion. For easy–hard retrieval analysis, we separate candidates into easier and harder subsets according to the retrieval protocol used in our experiments. For analogy-style retrieval, we treat a retrieved neighbor as *same-label* if it shares the exact gold relation label, and as *analogous* if it does not but still instantiates a human-interpretable structurally similar relation.

5.4 Implementation Details

All models use the same frozen contextual backbone and relation head, so the main difference lies in the training objective. We compare four variants—encoder-only, contrastive with regularization, R-JEPA (prediction-only), and full R-JEPA—and report all main results averaged over three random seeds. Additional details on the encoder setup, predictive objective, qualitative evaluation, and context-ablation protocol are provided in Appendix B.

6 Results

6.1 Relation Retrieval and Internal Structure on Re-DocRED

Table 1 reports the main results on Re-DocRED over three random seeds. The main pattern is a geometry split across objectives. Full R-JEPA achieves the best retrieval metrics, including MRR, Recall@1, and Recall@5, whereas prediction-only training yields by far the strongest ARI and probe accuracy, indicating stronger internal structure. Figure 2 makes this split clear: the strongest retrieval model and the strongest structure-oriented model are not the same system. Retrieval quality and internal organization are therefore distinct properties of relation space.

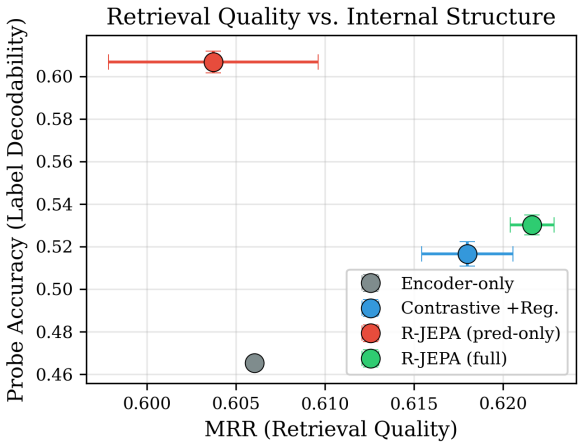

Figure 2: Retrieval quality versus internal label structure. The x-axis shows MRR and the y-axis shows probe accuracy. Full R-JEPA occupies the strongest retrieval region, whereas R-JEPA (pred-only) occupies the strongest structure-oriented region.

The retrieval advantage of full R-JEPA over the strongest discriminative baseline is modest in absolute magnitude, but it is consistent across three random seeds. The structural advantage of prediction-only training is much larger relative to seed variance, making it the most stable pattern in our experiments.

6.2 Easy and Hard Retrieval

The main-results table shows different overall geometries. We next ask whether this split is uniform across retrieval regimes or differs between easier and harder cases.

Table 2 separates retrieval into easy and hard subsets. Full R-JEPA performs best overall and on the easy subset. Prediction-only training performs best on hard MRR. The contrastive baseline remains strongest on hard Recall@5.

Method	Overall MRR	Easy MRR	Hard MRR	Easy R@5	Hard R@5
Encoder-only	0.6065	0.6203	0.4663	0.9163	0.6242
Contrastive + Reg.	0.6185	0.6335	0.4657	0.9251	**0.6532**
R-JEPA (pred-only)	0.6108	0.6240	**0.4767**	0.9152	0.6421
R-JEPA (full)	**0.6209**	**0.6365**	0.4622	**0.9277**	0.6309

Table 2: Easy and hard retrieval on Re-DocRED. Full R-JEPA is strongest overall, while prediction-only training is strongest on hard MRR.

Method	Mean Silhouette	Labels	With Sub-structure
Encoder-only	0.0495	93	6
Contrastive + Reg.	0.0520	93	6
R-JEPA (pred-only)	**0.3318**	93	**93**
R-JEPA (full)	0.0584	93	9

Table 3: Aggregate label-conditioned substructure statistics on Re-DocRED. Prediction-only training induces markedly stronger within-label organization.

These results suggest that the objectives shape neighborhood geometry differently: full R-JEPA improves global ranking, while prediction-only training is stronger on harder cases.

6.3 Within-Label Substructure

The easy–hard breakdown suggests that the objectives differ not only in aggregate performance but also in local neighborhood structure. We next test this directly by asking whether the learned spaces contain coherent organization within a single gold relation label.

For each high-frequency relation label, we cluster only instances of that label and compute aggregate statistics. Table 3 shows a large difference. Prediction-only training yields much larger silhouette values than the other methods and satisfies our current substructure criterion for nearly all analyzed labels. This is the strongest evidence in our experiments that predictive training can reveal latent organization inside a fixed label.

A second question is whether this pattern is merely geometric or also interpretable. Table 4 addresses that question. Under prediction-only training, labels such as *member of*, *educated at*, and *located in* split into human-interpretable subgroups. In *member of*, for example, the induced clusters separate country-to-organization cases, person-to-party or professional-organization cases, and person-to-band cases.

We therefore interpret prediction-only training as exposing latent within-label organization rather than merely inflating a clustering metric. One concern is that the model may simply over-partition

Relation label	Sil.	Interpretable subgroups
member of (P463)	0.52	countries→IGOs; people→parties/professional groups; people→bands
educated at (P69)	0.37	university education; specialized schools
located in (P131)	0.41	institutions→regions; geographic locations→countries/states

Table 4: Examples of human-interpretable within-label substructure under R-JEPA (prediction-only). Detailed exemplars are provided in the appendix.

labels without semantic value. However, the qualitative subgroup analyses show that the resulting partitions are often interpretable. We still do not treat these clusters as proof of a universal hidden ontology; stronger semantic claims would require additional human validation (Wang et al., 2025; Jamal et al., 2025).

6.4 Context Dependence

A natural alternative explanation for the preceding results is that predictive training simply makes better use of document context. We test this directly by varying the amount and quality of available context while holding the rest of the pipeline fixed. Contrary to our initial expectation, context reduction has very little effect on retrieval in the current frozen-span setup. As shown in Figure 3, the difference between full and single-sentence conditions remains below 0.5% in MRR for all methods. This indicates that the main differences among objectives are not primarily explained by superior document-context utilization under the current encoder design.

This result is still informative. In the present system, the gains of predictive training appear to arise mainly from objective-level reshaping of relation space on top of fixed local representations. Stronger context-sensitive effects may require a different encoder or a model that aggregates document evidence more directly.

6.5 Analogy-style Retrieval as Downstream Validation

If the geometry differences are meaningful rather than merely internal, they should also affect the kinds of neighbors each model retrieves beyond exact label matching. We therefore examine analogy-style retrieval as a downstream validation of the learned relation space.

Table 5 summarizes a qualitative evaluation over 50 retrieved neighbors. Aggregate differences are

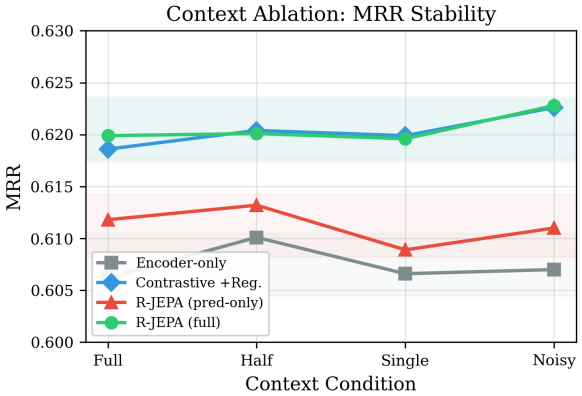

Figure 3: Context ablation visualized as MRR stability across four context conditions. All methods remain highly stable, supporting the interpretation that the main gains arise from objective-level reshaping of relation space rather than stronger document-context utilization.

Model	Same label	Analogous	Different	Same+Analog
R-JEPA (full)	20	15	15	35/50 (70%)
R-JEPA (pred-only)	24	12	14	**36/50 (72%)**
Contrastive + Reg.	21	13	16	34/50 (68%)

Table 5: Qualitative analogy-style retrieval summary. Aggregate differences are modest, but the types of retrieved neighbors differ across methods.

modest: all three learned models retrieve a comparable number of same-label or analogous neighbors.

The more important difference lies in the type of neighbors each model returns. Prediction-only training retrieves the highest number of exact same-label neighbors, which is consistent with its stronger within-label structure. Full R-JEPA retrieves slightly more cross-label analogous neighbors, which suggests that its smoother geometry better captures structural similarity across labels in the sense of relational alignment rather than exact class identity (Turney, 2006; Gentner, 1983; Gentner and Markman, 1997).

7 Discussion

The main lesson of this study is that predictive relation learning should not be evaluated through a single notion of representation quality. On Re-DocRED, different objectives induce different relation geometries. Full R-JEPA is strongest for retrieval, while prediction-only training yields much stronger within-label organization.

7.1 What Predictive Learning Improves

Our first research question asked whether predictive learning improves relation retrieval. The answer is cautiously positive. On Re-DocRED, full R-JEPA achieves the best overall retrieval performance among the compared methods. The margin over the strongest discriminative baseline is modest, but it is consistent across three random seeds.

This matters because retrieval is the most direct test of whether the learned relation space is practically useful. Full R-JEPA does not dominate every alternative by a large margin, but it shows that predictive training can match or slightly exceed strong discriminative baselines in this setting.

The retrieval gains are modest. Our claim is therefore not that predictive training yields a uniformly dominant retrieval model, but that it remains competitive on retrieval while inducing a different and analytically useful relation geometry.

7.2 What Kind of Relation Space Predictive Training Learns

Our second research question asked what kind of relation space predictive training produces. Here the answer is clearer. Prediction-only training yields substantially better ARI, probe accuracy, and label-conditioned substructure than the other methods. This structural advantage is large relative to seed variance, making it the most stable pattern in the paper.

The qualitative analyses strengthen this conclusion. In labels such as *member of*, *educated at*, and *located in*, the induced clusters are not merely separable but also interpretable. For example, *member of* separates into country-to-organization, person-to-party or professional-organization, and person-to-band relations.

Another concern is that prediction-only training may simply overcluster the data. Our interpretation is narrower. The point is not that every induced subgroup corresponds to a valid new ontology, but that the resulting partitions are far more structured and often more interpretable than those produced by the baselines.

The combined implication is important. Full R-JEPA yields the best retrieval, while prediction-only training yields the best within-label organization. Retrieval-oriented geometry and label-aligned structure should therefore be treated as distinct properties of relation space.

7.3 Why Context Dependence Still Matters

Our third research question concerned context dependence. Conceptually, this remains central.

Document-level relation extraction is attractive because some relations are not identifiable from a local pair alone; they must be inferred from partial, distributed, or noisy evidence (Yao et al., 2019; Christopoulou et al., 2019; Zeng et al., 2020; Xu et al., 2021; Zhou et al., 2021; Zhang et al., 2025a).

Empirically, however, our context-ablation results show almost no degradation for any method under reduced or noisy context. In the present frozen-span setup, the observed gains cannot be attributed mainly to superior context utilization. Instead, they arise mostly from objective-level reshaping of relation space.

This interpretation sharpens the paper. In the current system, R-JEPA should be understood primarily as a predictive relation-space objective on top of fixed local representations. Stronger context-sensitive gains may require a different encoder design or a model that aggregates document evidence more directly.

7.4 What Generalization Looks Like

Our fourth research question concerned downstream generalization. We are interested in analogy-style neighborhoods because they are a natural downstream use of a learned relation space. The current results do not solve analogy retrieval as a standalone task, but they do show that different predictive variants support different forms of generalization.

Prediction-only training tends to preserve tighter same-label neighborhoods, which is consistent with its stronger within-label structure. Full R-JEPA, by contrast, more often retrieves cross-label but structurally related examples, which suggests that it induces a smoother relation space that supports broader relational generalization in the sense of structural similarity rather than exact label identity (Turney, 2006; Gentner, 1983; Gentner and Markman, 1997; Wang et al., 2025).

Aggregate analogy-style counts are similar across models. The more important point is that the learned neighborhoods differ in meaningful ways. This supports the broader claim that predictive relation learning can organize relation space beyond exact label identity.

7.5 Scope of the Claims

The contribution of this paper is best understood at three levels. First, we introduce a methodological reframing: document-level relation learning can be formulated as context-to-target prediction in embedding space rather than direct label prediction. Second, we provide an empirical finding: on Re-DocRED, predictive training does not yield a single best relation geometry, but instead separates into a retrieval-oriented variant (full R-JEPA) and a structure-oriented variant (prediction-only R-JEPA). Third, we make a conceptual claim: retrieval quality, within-label structure, and analogy-style neighborhood formation are related but distinct properties of relation space, and they should not be collapsed into a single notion of representation quality.

The boundaries of the current study are also clear. We do not claim that predictive relation learning is always stronger than discriminative training. We do not claim that within-label clusters are automatically meaningful sub-relations. We do not claim that analogy retrieval has already been solved. Our claim is narrower: predictive relation learning is a viable framework for document-level relation modeling, and it reveals aspects of relation-space organization that standard discriminative evaluation alone can miss.

8 Conclusion

We presented R-JEPA, a predictive framework for document-level relation representation learning that formulates relation learning as context-to-target prediction in embedding space rather than direct label prediction. Our central finding is that predictive training does not induce a single best relation geometry. Instead, it improves different properties of relation space in different ways: full R-JEPA performs best for retrieval, whereas prediction-only training yields much stronger within-label structure and more interpretable subgroup formation.

These findings suggest that relation-space quality should not be evaluated through a single end metric alone. Retrieval quality, internal label-aligned organization, and analogy-style neighborhood formation are related but distinct properties, and different objectives optimize them differently. More broadly, our results support predictive relation learning as a useful framework for document-level relation modeling and motivate future work on stronger context-sensitive encoders, broader benchmark coverage, and more standardized evaluation of latent relational structure.

9 Limitations

This study does not claim discovery of objectively correct novel relation categories. It studies whether predictive learning induces a useful relation space under practical evaluation protocols. Several limitations remain.

First, some conclusions depend on the current encoder design. Because the encoder is frozen, the observed gains appear to arise mainly from objective-level reshaping of relation space rather than from stronger context use. Second, within-label substructure analyses are still exploratory and require qualitative or human validation to support stronger semantic claims. Third, the differences among the strongest retrieval systems are modest, although the main ranking pattern is consistent across three random seeds. A fuller statistical analysis would further strengthen the case, especially for close comparisons between full R-JEPA and the strongest discriminative baseline. Fourth, our analogy-style retrieval analysis remains qualitative and depends on human interpretation of structural similarity beyond exact label matching. A more standardized annotation protocol would strengthen this part of the evaluation. Finally, we evaluate only one benchmark, because our goal is a controlled analysis rather than broad benchmark coverage. Additional datasets would help determine how broadly the observed geometry split generalizes.

10 Ethics Statement

This work studies predictive representation learning for relation understanding. Potential risks include overclaiming the meaning of emergent clusters, amplifying biases present in pretrained encoders and relation datasets, and encouraging users to treat latent structure as ground truth without sufficient evidence. We therefore keep our claims narrow. We focus on retrieval quality, internal organization, and transfer rather than on strong ontological claims.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905. Association for Computational Linguistics.

Adrien Bardes, Jean Ponce, and Yann LeCun. 2022. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607.

Shabnam Choudhury, Yash Salunkhe, Sarthak Mehrotra, and Biplab Banerjee. 2025. REJEPA: A novel joint-embedding predictive architecture for efficient remote sensing image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936. Association for Computational Linguistics.

Julien Delaunay, Hanh Thi Hong Tran, Carlos-Emiliano González-Gallardo, Georgeta Bordea, Nicolas Sidere, and Antoine Doucet. 2023. A comprehensive survey of document-level relation extraction (2016–2023). *arXiv preprint arXiv:2309.16396*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.

Dedre Gentner and Arthur B. Markman. 1997. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45–56.

Ridong Han, Tao Peng, Benyou Wang, Lu Liu, Prayag Tiwari, and Xiang Wan. 2024. Document-level relation extraction with relation correlations. *Neural Networks*, 171:14–24.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification

dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809. Association for Computational Linguistics.

Hai Huang, Yann LeCun, and Randall Balestriero. 2026. LLM-JEPA: Large language models meet joint embedding predictive architectures. In *The Thirteenth International Conference on Learning Representations*.

Rebii Jamal, Mounir Ourekouch, and Mohammed Erradi. 2025. UOREX: Towards uncertainty-aware open relation extraction. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. DIRT – discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328.

Gaetano Rossiello, Alfio Gliozzo, Robert Farrell, Nicolas Fauceglia, and Michael Glass. 2019. Learning relational representations by analogy using hierarchical siamese networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1766–1776. Association for Computational Linguistics.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting DocRED – addressing the false negative problem in relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8525–8536. Association for Computational Linguistics.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Asahi Ushio, Jose Camacho-Collados, and Steven Schockaert. 2021. Distilling relation embeddings from pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9042–9054. Association for Computational Linguistics.

Asahi Ushio, Jose Camacho-Collados, and Steven Schockaert. 2025. RelBERT: Embedding relations with language models. *Artificial Intelligence*, 347:104359.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Qing Wang, Yuepei Li, Qiao Qiao, Kang Zhou, and Qi Li. 2025. Towards a more generalized approach in open relation extraction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6343–6354. Association for Computational Linguistics.

Qing Wang, Kang Zhou, Qiao Qiao, Yuepei Li, and Qi Li. 2023. Improving unsupervised relation extraction by augmenting diverse sentence pairs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12136–12147. Association for Computational Linguistics.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14149–14157.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777. Association for Computational Linguistics.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640. Association for Computational Linguistics.

Fu Zhang, Yi Yan, and Jingwei Cheng. 2025a. Document-level relation extraction with global relations and entity pair reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19556–19567. Association for Computational Linguistics.

Fu Zhang, Hongsen Yu, Jingwei Cheng, and Huangming Xu. 2025b. Entity pair-guided relation summarization and retrieval in LLMs for document-level relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4022–4037. Association for Computational Linguistics.

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021. Document-level relation extraction as semantic segmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3999–4006.

Jiaming Zhao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Actively supervised clustering for open relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4939–4954. Association for Computational Linguistics.

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14612–14620.

A Dataset Details

A.1 Re-DocRED Overview

We use **Re-DocRED** (Tan et al., 2022), a revised version of DocRED (Yao et al., 2019). The original DocRED is a large-scale, document-level relation extraction dataset constructed from Wikipedia and Wikidata. Each document contains multiple entities (identified through coreference) and multiple relation labels expressed across sentence boundaries. Re-DocRED revises the training and development annotations of DocRED to reduce the annotation incompleteness problem: many valid relation instances in the original dataset were not annotated, producing false negatives that can distort both training and evaluation. We use the revised training split (`train_revised`) throughout all experiments.

A.2 Dataset Statistics

Table 6 summarizes the key statistics of the dataset as used in our experiments.

Property	Value
Dataset variant	Re-DocRED
Split used	`train_revised`
Total entity pairs	85,932
Relation types	96
Entity representation	span_mean_boundary
Entity embedding dimension	2,304
Projected entity dimension	768
Relation embedding dimension	128
Contextual backbone	BERT-base-uncased
Backbone parameters	110M (frozen)
Max sequence length	512 subword tokens

Table 6: Dataset and representation statistics for the Re-DocRED setup used in all experiments.

A.3 Relation Type Distribution

Re-DocRED contains 96 Wikidata relation types. The distribution is highly skewed. The five most frequent relation types—*located in the administrative territorial entity* (P131), *country* (P17), *instance of* (P31), *country of citizenship* (P27), and *occupation* (P106)—together account for over 60% of all entity pairs. The median label frequency is approximately 302 pairs. The least frequent relation types have fewer than 10 instances. This skewed distribution motivates the easy–hard retrieval analysis in Section 7.2 and Appendix E.2, where "easy" labels are those with above-median frequency and "hard" labels are those below the median.

A.4 Document Structure

Each Re-DocRED document consists of multiple tokenized sentences. Entity mentions are annotated with span positions (word-level start and end indices within a sentence) and entity types (PER, ORG, LOC, NUM, TIME, MISC). A single entity may have multiple mentions across different sentences in the same document, and entity identity is resolved through coreference annotation in the original dataset. Relation labels are specified as `(head_entity, tail_entity, relation_type)` triples, where the head and tail refer to entity indices within the document's vertex set.

A.5 Why Re-DocRED Instead of DocRED

We use Re-DocRED rather than the original DocRED because our evaluation goes beyond classification accuracy. In retrieval and within-label substructure analysis, false negatives are particularly harmful: if two instances share the same true relation but only one is annotated, the retrieval evaluation will penalize the model for correctly placing them near each other. The revised annotations in Re-DocRED reduce this problem, making retrieval-based and structure-based evaluation more reliable.

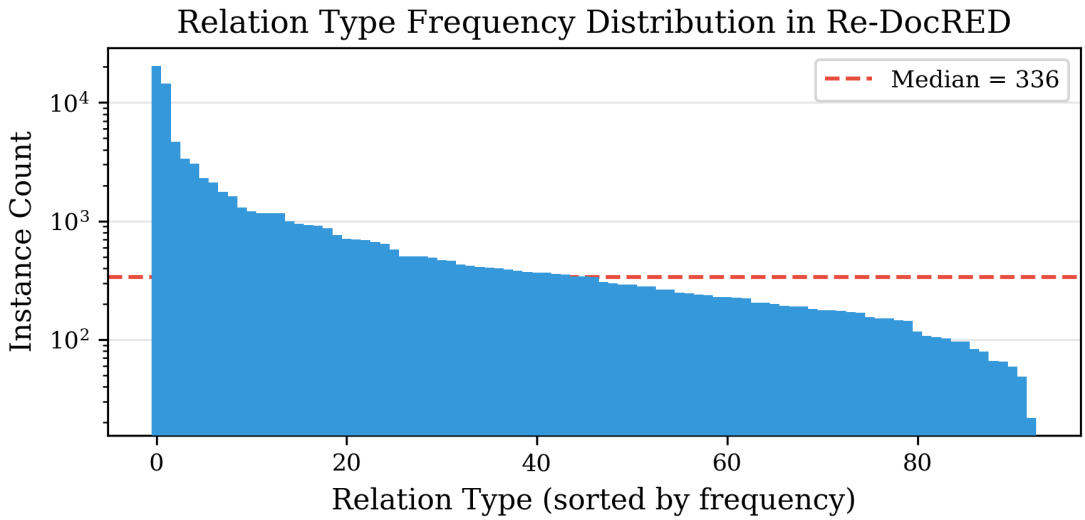

Figure 4: Frequency distribution of relation types in Re-DocRED, sorted by instance count. The distribution is strongly long-tailed, with a small number of very frequent relation types and many low-frequency types. The dashed line marks the median frequency used for the easy–hard split.

B Implementation Details

B.1 Entity Encoding Pipeline

Entity representations are computed once and cached to disk before training. The pipeline proceeds as follows.

Step 1: Document tokenization. Each document's sentences are concatenated into a single word sequence. The BERT tokenizer (`bert-base-uncased`) tokenizes this sequence using the `is_split_into_words=True` mode, which preserves the alignment between original word indices and BERT subword tokens. The tokenizer's `word_ids()` method provides the mapping from each subword token position back to its original word index.

Step 2: BERT encoding. The tokenized document is passed through a frozen BERT-base-uncased model (Devlin et al., 2019). No gradients flow through the backbone at any point during dataset construction or training. The last hidden layer produces a sequence of 768-dimensional vectors, one per subword token.

Step 3: Span extraction. For each entity mention, the word-level span positions (relative to the mention's sentence) are converted to subword token positions using the word-to-subword alignment from Step 1. The entity mention representation is then computed as:

$$\mathbf{v}_{\text{mention}} = [\text{mean}(\mathbf{H}_{s:e});\ \mathbf{H}_s;\ \mathbf{H}_{e-1}] \tag{1}$$

where $\mathbf{H}_{s:e}$ denotes the hidden states for subword tokens in the span $[s, e)$, $\mathbf{H}_s$ is the first token of the span, and $\mathbf{H}_{e-1}$ is the last. This yields a 2,304-dimensional vector (768×3) that captures both the average span semantics and boundary information.

Step 4: Multi-mention aggregation. When an entity has multiple mentions in the document, the final entity representation is the mean of all mention representations:

$$\mathbf{e} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathbf{v}_m \tag{2}$$

where $\mathcal{M}$ is the set of mentions for that entity.

Step 5: Caching. The resulting entity-pair tensors (`pairs_ei`, `pairs_ej`), relation labels, and metadata are saved to disk as a PyTorch cache file. Subsequent training runs load this cache directly, avoiding repeated BERT inference. Cache filenames encode the dataset variant, split, entity representation type, and context mode (e.g., `re-docred_train_revised_span_mean_boundary.pt`).

Alternative entity representation. We also support a simpler `span_mean` mode that uses only the mean-pooled span hidden states (768-dimensional), omitting the boundary tokens. In this mode, no entity projection is needed since the entity dimension already matches the pair encoder's expected input size.

B.2 Model Architecture

Figure 5 illustrates the full R-JEPA architecture. All architectural hyperparameters are listed in Table 7.

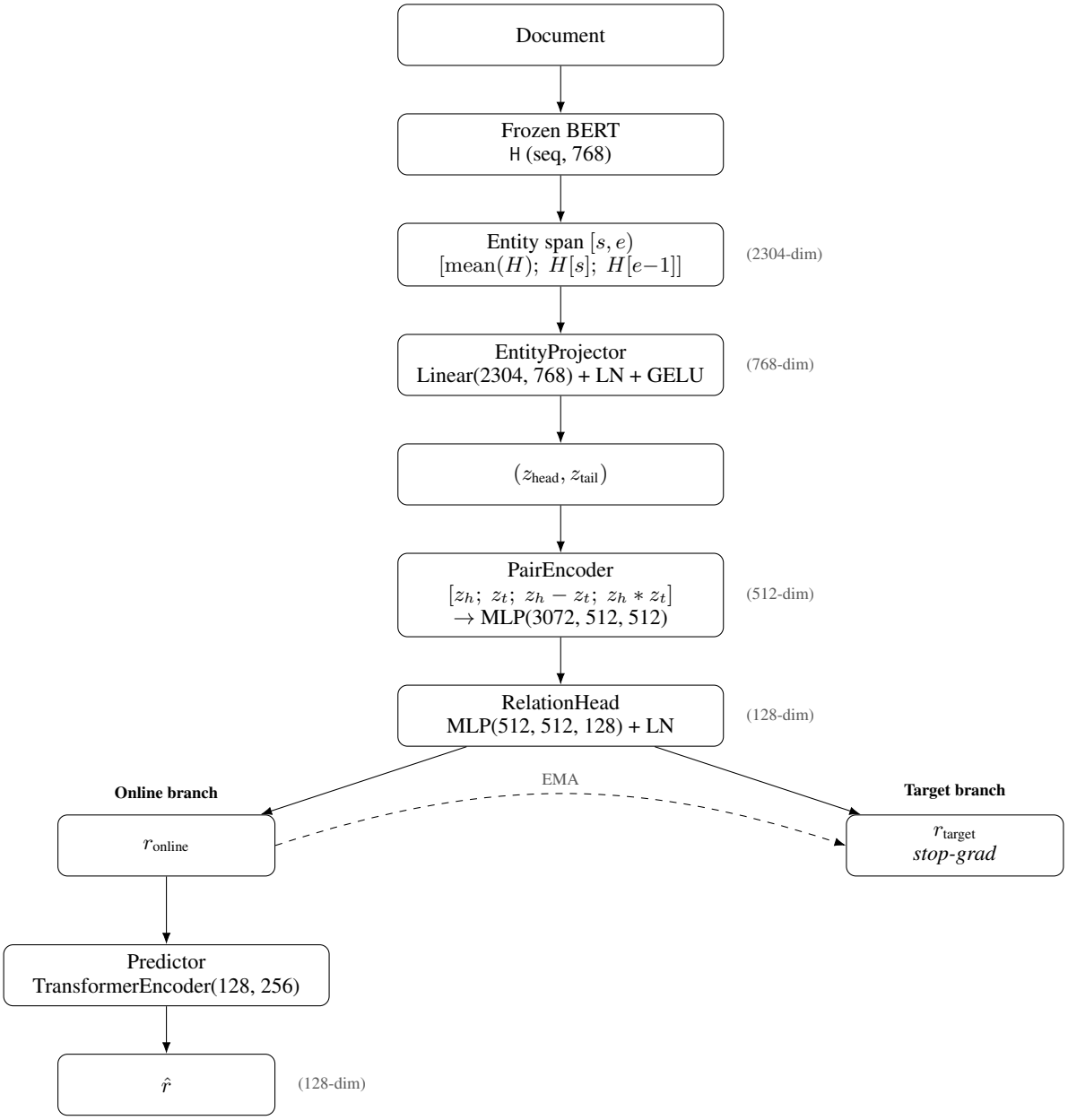

Figure 5: R-JEPA architecture overview. The document is encoded by a frozen BERT backbone, entity spans are converted into pair representations, and the relation head maps them into relation space. The online branch predicts a target relation embedding, while the target branch is updated by EMA and uses stop-gradient.

Entity Projector. When the raw entity dimension (2,304 for `span_mean_boundary`) exceeds the pair encoder's expected input size, a learned projection layer maps entity representations to the working dimension:

$$\mathbf{z} = \text{GELU}(\text{LN}(\mathbf{W}\mathbf{e} + \mathbf{b})) \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{768\times 2304}$ and LN denotes layer normalization. This projector is part of the online branch and is trained end-to-end with the relation-learning objective. Its parameters are included in the EMA update to the target branch.

Pair Encoder. The pair encoder takes two projected entity representations and produces a pair-level feature vector. Following prior work on relation representation (Baldini Soares et al., 2019), the input is a four-way concatenation:

$$\mathbf{u} = [\mathbf{z}_h;\ \mathbf{z}_t;\ \mathbf{z}_h - \mathbf{z}_t;\ \mathbf{z}_h \odot \mathbf{z}_t] \tag{4}$$

where $\odot$ denotes element-wise multiplication. This 3,072-dimensional input (768×4) is processed by a 2-layer MLP with hidden dimension 512, using LayerNorm and GELU activations after each layer, with dropout rate 0.1.

Relation Head. The relation head maps the 512-dimensional pair representation to a 128-dimensional relation embedding. It consists of a 2-layer MLP with LayerNorm applied to the final output. This normalization ensures that all relation embeddings lie on a similar scale, which is important for cosine-similarity-based retrieval.

Predictor. The predictor takes a set of k context relation embeddings and produces a prediction of the target relation embedding. We use an attention-based predictor consisting of:

1. A linear projection from the relation dimension (128) to the predictor hidden dimension (256).
2. A 2-layer Transformer encoder with 4 attention heads, feedforward dimension 512, GELU activation, pre-norm architecture, and dropout 0.1.
3. Mean pooling across the k context positions.
4. A LayerNorm and linear projection back to the relation dimension (128).

The predictor is intentionally smaller than the encoder, following the JEPA design principle that the predictor should be a bottleneck that forces the encoder to learn broadly useful representations rather than memorizing context-specific patterns (Assran et al., 2023).

Target Branch. The target encoder, relation head, and entity projector are copies of their online counterparts, updated via exponential moving average (EMA) with momentum $m = 0.996$:

$$\theta_{\text{tar}} \leftarrow 0.996 \cdot \theta_{\text{tar}} + 0.004 \cdot \theta_{\text{online}} \tag{5}$$

The target branch uses stop-gradient: no gradients flow through the target computation. This asymmetry prevents representation collapse (Assran et al., 2023).

B.3 Loss Functions

We use four loss terms, combined with different weights depending on the training variant.

Prediction loss. The primary JEPA objective is mean squared error between the predicted and target relation embeddings:

$$\mathcal{L}_{\text{pred}} = \frac{1}{d}\sum_{i=1}^{d}(\hat{r}_i - r_i^{\text{tar}})^2 \tag{6}$$

Component	Parameter	Value
Entity Projector	Input dim	2,304
	Output dim	768
Pair Encoder	Input dim	3,072
	Hidden dim	512
	Layers	2
	Dropout	0.1
Relation Head	Input dim	512
	Output dim	128
	Layers	2
Predictor	Hidden dim	256
	Attn. heads	4
	Transformer layers	2
	FF dim	512
EMA	Momentum	0.996

Table 7: Architectural hyperparameters for all R-JEPA components.

Variance loss. Following VICReg (Bardes et al., 2022), the variance regularizer prevents dimensional collapse by encouraging each embedding dimension to maintain a minimum standard deviation:

$$\mathcal{L}_{\text{var}} = \frac{1}{d}\sum_{j=1}^{d} \max\left(0, \gamma - \sqrt{\text{Var}[r_j] + \epsilon}\right) \tag{7}$$

where $\gamma = 1.0$ is the target standard deviation and $\epsilon = 10^{-4}$ prevents numerical instability.

Covariance loss. The covariance regularizer decorrelates embedding dimensions to prevent redundancy:

$$\mathcal{L}_{\text{cov}} = \frac{1}{d}\sum_{i \neq j} \text{Cov}[r_i, r_j]^2 \tag{8}$$

where the covariance matrix is computed from centered embeddings within each batch.

Discriminative loss. An optional InfoNCE-style contrastive loss (van den Oord et al., 2018) provides explicit positive–negative discrimination:

$$\mathcal{L}_{\text{disc}} = -\log \frac{\exp(\cos(\mathbf{a}, \mathbf{p})/\tau)}{\exp(\cos(\mathbf{a}, \mathbf{p})/\tau) + \sum_n \exp(\cos(\mathbf{a}, \mathbf{n})/\tau)} \tag{9}$$

where $\mathbf{a}$ is the anchor (predicted relation for JEPA, mean context for contrastive), $\mathbf{p}$ is the positive target, $\mathbf{n}$ are hard negatives, and $\tau = 0.07$ is the temperature.

B.4 Training Variants

Table 8 specifies the exact loss configuration for each compared model. All four variants share the same encoder architecture, entity projector, and relation head. They differ only in which loss terms are active and whether a predictor and target branch are used.

Variant	Mode	λ_{pred}	λ_{var}	λ_{cov}	λ_{disc}	Predictor
Encoder-only	encoder	–	1.0	1.0	–	No
Contrastive+Reg.	contrastive	–	1.0	0.04	1.0	No
R-JEPA (pred-only)	jepa	1.0	0.0	0.0	0.0	Yes
R-JEPA (full)	jepa	1.0	1.0	0.04	0.1	Yes

Table 8: Loss weight configuration for each training variant. "–" indicates the term is not applicable to that training mode.

Encoder-only. Uses only variance and covariance regularization. There is no contrastive or predictive signal. This baseline tests whether geometric regularization alone can organize relation space.

Contrastive + Regularization. Uses InfoNCE contrastive loss with VICReg-style regularization. The anchor is the mean-pooled context relation embedding, and the positive is the target relation embedding from the online branch. Hard negatives are sampled from different relation types within the same coarse category. There is no predictor or target branch.

R-JEPA (prediction-only). Uses only the MSE prediction loss. No variance, covariance, or discriminative regularization is applied. This isolates the effect of the predictive objective itself.

R-JEPA (full). Combines the prediction loss with all three regularization terms and a discriminative term. This tests whether the combination of predictive and regularization objectives yields better retrieval than either alone.

B.5 Episode Sampling

Training proceeds in episodes. Each episode consists of a context set, a target, and a set of hard negatives.

Context and target construction. For each episode, a relation type y is sampled uniformly from the set of eligible types (those with at least $k+1$ instances, where k is the context size). Then k context instances and 1 target instance are sampled without replacement from all instances with label y. In our experiments, $k = 4$.

Hard negative sampling. For each episode, 8 hard negatives are sampled. Hard negatives are instances whose relation type differs from the context/target type. This makes the contrastive and discriminative objectives more challenging than random negative sampling.

Episode batching. Episodes are batched into groups of 32. Each batch produces tensors of shape $(32, 4, D)$ for context entities, $(32, D)$ for target entities, and $(32, 8, D)$ for negative entities, where D is the entity embedding dimension.

B.6 Training Configuration

Table 9 summarizes the training hyperparameters used for all experiments.

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	1×10^{-5}
Gradient clipping (max norm)	1.0
Batch size (episodes)	32
Episodes per epoch	512
Total epochs	20
Warmup epochs	10
LR schedule (warmup)	Linear (0.01× to 1×)
LR schedule (post-warmup)	Cosine annealing
EMA momentum	0.996
Context size k	4
Hard negatives per episode	8
Evaluation frequency	Every 5 epochs
Random seeds	{42, 123, 7}

Table 9: Training hyperparameters for all Re-DocRED experiments.

Optimization. We use AdamW with learning rate 3×10^{-4} and weight decay 10^{-5}. Only the online branch parameters are optimized (entity projector, online encoder, online relation head, and predictor). The target branch is updated exclusively via EMA.

Learning rate schedule. The first 10 epochs use a linear warmup from $0.01 \times$ lr to $1.0 \times$ lr. The remaining 10 epochs use cosine annealing down to 0.

Gradient clipping. Gradients are clipped to a maximum norm of 1.0 before each optimizer step. This prevents training instability, particularly in the early epochs when the predictor has not yet learned useful predictions.

Model selection. The best checkpoint is selected based on MRR on the training set, evaluated every 5 epochs (at epochs 5, 10, 15, and 20). We use training-set evaluation rather than a held-out validation set because our goal is to study the properties of the learned relation space rather than to optimize generalization to unseen documents. This design choice means that reported metrics reflect the quality of the relation space that the model has learned, not its ability to generalize to new documents.

B.7 Computational Cost

Table 10 reports the wall-clock training time for each variant on a single GPU.

Variant	Training time (seconds)
Encoder-only	∼580
Contrastive + Reg.	∼690
R-JEPA (pred-only)	∼930
R-JEPA (full)	∼790

Table 10: Approximate wall-clock training time per run (20 epochs) on a single NVIDIA GPU. The JEPA variants are slower due to the predictor forward pass and EMA update.

Entity encoding with frozen BERT is a one-time cost (approximately 15–20 minutes for the full `train_revised` split) and is cached to disk. All subsequent training runs load the cached entity embeddings directly.

C Evaluation Details

C.1 Relation Retrieval Protocol

Retrieval evaluation proceeds as follows. After training, all entity pairs in the dataset are encoded through the online branch to produce relation embeddings $\{r_1, \ldots, r_N\}$ with $N = 85{,}932$. To manage computational cost, we subsample 5,000 query instances uniformly at random (with a fixed seed of 42 for reproducibility).

For each query r_q:

1. Compute cosine similarity $\cos(r_q, r_i)$ for all $i \neq q$.
2. Rank all candidates by decreasing similarity.
3. Find the rank of the first candidate that shares the same relation type as the query.
4. Compute MRR as $1/\text{rank}$ (or 0 if no match is found among all candidates).
5. Compute Recall@k as 1 if any of the top-k candidates shares the query's label, 0 otherwise.

Final metrics are averaged over all 5,000 queries. We report MRR, Recall@1, Recall@5, and Recall@10.

C.2 Easy–Hard Retrieval Split

The easy–hard split is defined by label frequency. We compute the number of instances per relation type and find the median frequency (approximately 302). Queries whose relation type has above-median frequency are classified as "easy"; those with below-median frequency are "hard." This split reflects the intuition that retrieval is easier when the query's relation type has many candidates in the dataset and harder when it has few. Out of 5,000 queries, approximately 4,553 are easy and 447 are hard.

C.3 Clustering Metrics

We evaluate label-aligned clustering using K-means with $K = 96$ (matching the number of relation types). We report:

- **Adjusted Rand Index (ARI):** Measures agreement between predicted clusters and gold labels, adjusted for chance. Ranges from -1 to 1; higher is better.
- **Normalized Mutual Information (NMI):** Measures mutual information between cluster assignments and gold labels, normalized to $[0, 1]$.
- **Silhouette coefficient:** Measures how similar each point is to its own cluster versus the nearest other cluster. Computed on a subsample of 5,000 instances.

K-means is run with 10 random initializations (`n_init=10`) and a fixed random seed (42) for reproducibility.

C.4 Linear Probe

The linear probe evaluates how much gold-label information is linearly decodable from frozen relation embeddings. We split the data into 80% training and 20% test (using a fixed permutation with seed 42) and train a logistic regression classifier (`max_iter=1000`) on the frozen relation embeddings. We report test accuracy. This metric complements ARI by measuring supervised rather than unsupervised label recovery.

C.5 Within-Label Substructure Analysis

For each relation label with at least 20 instances, we:

1. Extract all relation embeddings belonging to that label.
2. Run K-means with $K \in \{2, 3, 4\}$, each with 5 random initializations.
3. Select the K that yields the highest silhouette score.
4. Report the silhouette score, cluster sizes, and representative exemplars (the 5 instances closest to each cluster centroid by cosine distance).

A label is said to exhibit "substructure" if its best silhouette score exceeds 0.1. This threshold is conservative; silhouette values above 0.1 generally indicate that at least some cluster separation is present, though not necessarily strong separation. We analyze 93 out of 96 relation types (those with $\geq$20 instances).

C.6 Analogy-style Retrieval Annotation

For the analogy-style retrieval analysis, we select 5 query instances from diverse relation types (P27, P69, P463, P131, P39) and retrieve the 10 nearest neighbors for each query. Each retrieved neighbor is manually annotated with one of three labels:

- **Same label:** The neighbor shares the exact gold relation type with the query.
- **Analogous:** The neighbor has a different gold relation type but instantiates a structurally similar relation, as defined by the analogy groups in Table 11.
- **Different:** The neighbor has a different relation type with no clear structural similarity to the query.

The analogy groups are defined a priori based on Wikidata relation semantics, not post hoc from the model's output. This reduces the risk of confirmation bias in the annotation.

Analogy group	Relation types (Wikidata PIDs)
Location / containment	P131, P17, P27, P19, P20, P276, P159, P551, P937, P740, P1376, P36
Membership / affiliation	P463, P54, P102, P108, P241
Temporal events	P569, P570, P571, P577, P580, P582, P585
Creative works	P50, P57, P58, P170, P162, P178
Family	P22, P25, P26, P40, P3373, P451
Part–whole	P527, P361, P150
Education	P69
Performance / production	P175, P161, P264, P272, P449
Position / role	P39, P6, P35, P488

Table 11: Analogy groups used for annotating cross-label structural similarity. Two relations are considered analogous if they belong to the same group. Groups are defined a priori from Wikidata relation semantics.

D Context Ablation Details

D.1 Context Modes

We define four context conditions that control how much document text is available when computing entity span representations. All conditions use the same frozen BERT backbone and the same entity span extraction procedure (Appendix B.1). The difference lies in which sentences are included when BERT processes the document.

- **Full:** All sentences in the document are included. This is the default condition.
- **Half:** All sentences containing entity mentions are kept, plus approximately 50% of the remaining sentences (sampled randomly with a fixed seed).
- **Single:** Only sentences that contain at least one entity mention are included. All other sentences are removed.
- **Noisy:** All sentences are included, plus 1–2 randomly selected sentences from other documents are injected at random positions.

Sentence-to-span alignment. When sentences are removed (half, single), the word-level entity positions must be re-mapped to account for the shortened document. We maintain a mapping from original sentence indices to kept sentence indices and adjust span positions accordingly. When sentences are injected (noisy), injected sentences are marked with a sentinel index (-1) so they are never confused with genuine entity-bearing sentences.

Separate caching. Each context mode produces a separate cached file. This ensures that different context conditions do not contaminate each other and that results are fully reproducible.

D.2 Full Context Ablation Results

Table 12 provides the complete context ablation results, including all metrics (not just MRR as in the main text).

D.3 Interpretation

The stability of all metrics across context conditions is remarkably consistent. Not only MRR but also ARI, NMI, and probe accuracy remain nearly unchanged. This pattern holds across all four training variants, suggesting that it is a property of the frozen BERT span encoding rather than of any particular training objective.

This finding has two implications. First, in the current setup, the frozen BERT encoder effectively compresses entity identity into local span information, and document-level context adds little additional signal. Second, the differences among training objectives (which are substantial on metrics like ARI and probe accuracy) arise entirely from how those objectives reshape the relation space on top of these fixed local features.

Method	Context	MRR	R@1	R@5	R@10	ARI	Probe
R-JEPA (full)	Full	0.6199	0.4222	0.9020	0.9390	0.033	0.524
	Half	0.6201	0.4242	0.9010	0.9382	0.033	0.528
	Single	0.6196	0.4226	0.9056	0.9384	0.034	0.521
	Noisy	0.6228	0.4252	0.9056	0.9430	0.034	0.518
Contrastive+Reg.	Full	0.6186	0.4216	0.9016	0.9382	0.032	0.509
	Half	0.6204	0.4244	0.9006	0.9362	0.030	0.512
	Single	0.6199	0.4240	0.9040	0.9398	0.032	0.522
	Noisy	0.6226	0.4274	0.9026	0.9384	0.032	0.519
R-JEPA (pred-only)	Full	0.6118	0.4134	0.8908	0.9332	0.073	0.612
	Half	0.6132	0.4164	0.8928	0.9348	0.071	0.614
	Single	0.6089	0.4134	0.8882	0.9300	0.069	0.614
	Noisy	0.6110	0.4142	0.8910	0.9300	0.075	0.608
Encoder-only	Full	0.6062	0.4112	0.8902	0.9288	0.023	0.467
	Half	0.6101	0.4160	0.8920	0.9296	0.026	0.466
	Single	0.6066	0.4116	0.8912	0.9266	0.024	0.472
	Noisy	0.6070	0.4102	0.8908	0.9258	0.024	0.462

Table 12: Complete context ablation results on Re-DocRED. All metrics remain stable across context conditions for all methods, confirming that differences among objectives arise from the training signal rather than from context utilization.

E Additional Results

E.1 Seed-Level Results

Table 13 reports the full per-seed results for all four methods, providing the raw data behind the summary statistics in Table 1.

Method	Seed	MRR	R@1	R@5	R@10	ARI	NMI	Probe
R-JEPA (full)	42	0.6199	0.4222	0.9020	0.9390	0.033	0.296	0.524
	123	0.6224	0.4242	0.9102	0.9434	0.035	0.314	0.535
	7	0.6226	0.4246	0.9056	0.9406	0.034	0.302	0.531
Contrastive+Reg.	42	0.6186	0.4216	0.9016	0.9382	0.032	0.283	0.509
	123	0.6208	0.4226	0.9058	0.9364	0.032	0.285	0.523
	7	0.6146	0.4208	0.8950	0.9340	0.032	0.274	0.518
R-JEPA (pred-only)	42	0.6118	0.4134	0.8908	0.9332	0.073	0.449	0.612
	123	0.5979	0.3968	0.8868	0.9276	0.086	0.431	0.600
	7	0.6015	0.3992	0.8926	0.9262	0.092	0.466	0.608
Encoder-only	42	0.6062	0.4112	0.8902	0.9288	0.023	0.238	0.467
	123	0.6060	0.4106	0.8842	0.9262	0.023	0.231	0.467
	7	0.6059	0.4086	0.8920	0.9294	0.022	0.226	0.462

Table 13: Per-seed results on Re-DocRED for all four methods. Each row represents one independent training run with the indicated random seed.

Variance analysis. Several patterns are visible in the seed-level data. Encoder-only has the smallest variance in MRR (std $= 0.0001$), which is expected since it applies minimal training signal and the relation space is dominated by the fixed entity projector and regularization. R-JEPA (pred-only) has the largest MRR variance (std $= 0.006$), suggesting that pure predictive training without regularization is more sensitive to initialization. However, pred-only's structural metrics (ARI, NMI, Probe) are consistently and substantially higher than all other methods across all seeds, making this the most robust finding in the paper.

E.2 Easy–Hard Retrieval: Full Results

Table 14 provides the complete easy–hard retrieval breakdown including Recall@1 and Recall@10 in addition to the MRR and Recall@5 shown in the main text.

Method	Overall MRR	Easy MRR	Hard MRR	Easy R@1	Hard R@1	Easy R@5	Hard R@5	Easy/Hard Count
Encoder-only	0.6065	0.6203	0.4663	0.4249	0.2752	0.9163	0.6242	4553/447
Contrastive+Reg.	0.6185	0.6335	0.4657	0.4384	0.2818	0.9251	**0.6532**	4553/447
R-JEPA (pred-only)	0.6108	0.6240	**0.4767**	0.4254	**0.3535**	0.9152	0.6421	4553/447
R-JEPA (full)	**0.6209**	**0.6365**	0.4622	**0.4378**	0.3535	**0.9277**	0.6309	4553/447

Table 14: Full easy–hard retrieval breakdown on Re-DocRED. Easy/hard labels are defined by median label frequency (≈302). Prediction-only training achieves the best hard MRR and hard Recall@1, while full R-JEPA is strongest on overall and easy metrics.

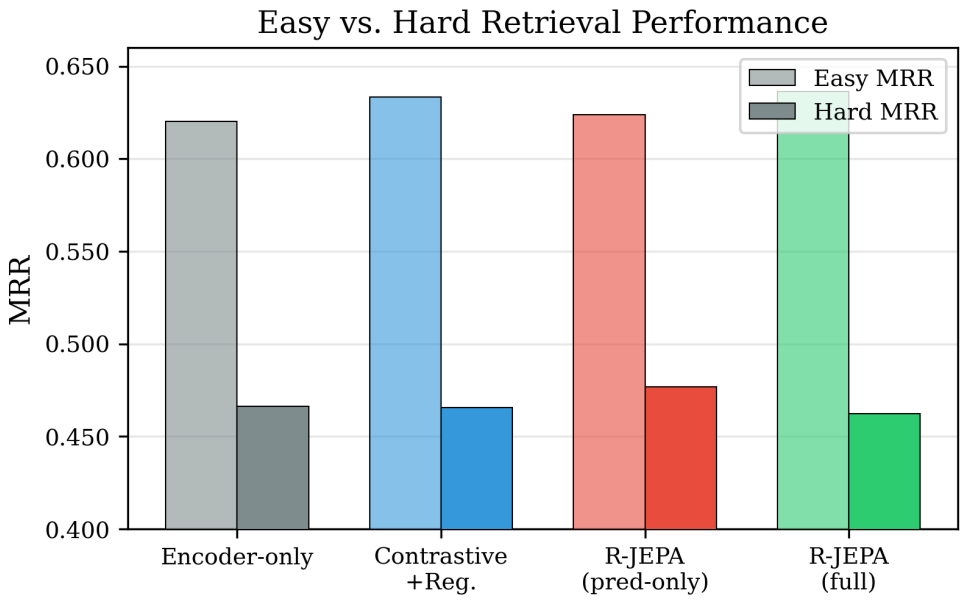

Figure 6: Easy versus hard retrieval performance. Prediction-only training performs best on the hard subset, while full R-JEPA is strongest on the easy subset. This supports the interpretation that stronger within-label compactness particularly benefits low-frequency relations.

Hard retrieval analysis. The hard subset (447 queries from low-frequency relation types) reveals the most interesting differences. R-JEPA (pred-only) achieves the highest hard MRR (0.4767) and ties for the best hard Recall@1 (0.3535). This is consistent with its stronger within-label structure: when a relation type has few instances, the tighter clustering induced by predictive training makes it easier to find the few matching candidates. Contrastive + Reg. achieves the best hard Recall@5 (0.6532), suggesting that its broader neighborhoods compensate for lower top-1 precision in the hard regime.

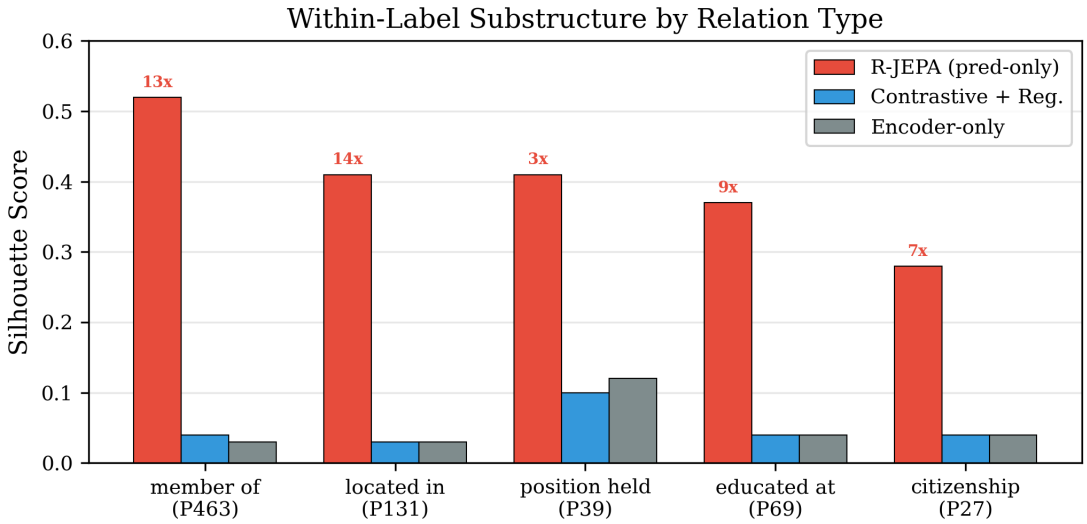

Figure 7: Within-label silhouette comparison on representative high-frequency relation labels. R-JEPA (pred-only) consistently yields much stronger within-label substructure than the encoder-only and contrastive baselines.

F Qualitative Examples

F.1 Within-Label Substructure: Model Comparison

Tables 15–17 provide detailed qualitative comparisons of within-label substructure across all three compared models. For each relation label, we show the cluster structure under R-JEPA (pred-only), Contrastive + Reg., and Encoder-only, including cluster sizes, silhouette scores, and representative exemplars.

F.1.1 Member of (P463)

Model	Sil.	Cluster	Pattern	Representative examples
R-JEPA (pred-only)	0.52	C0 (n=318)	country → intl. organization	(China, UN Security Council), (UK, UN Security Council), (USA, United Nations)
		C1 (n=472)	person → political/prof. org.	(Stefan Kaufmann, CDU), (Fabian Picardo, Gibraltar Socialist Labour), (Gaetano Cecere, National Sculpture Society)
		C2 (n=509)	person → music band	(Bert McCracken, The Used), (Justin Broadrick, Godflesh), (Annie Lennox, Eurythmics)
Contrastive +Reg.	0.04	mixed	weak partial separation	(Nigeria, UN), (Tony Fox Sales, Tin Machine), (Helen Suzman, Progressive Party)
Encoder-only	0.03	mixed	weak separation	(Russia, UN), (Stefan Kaufmann, CDU), (Tim Armstrong, Rancid)

Table 15: Within-label substructure comparison for *member of* (P463). R-JEPA (pred-only) discovers three clean, human-interpretable subgroups with a silhouette score 13× higher than the baselines.

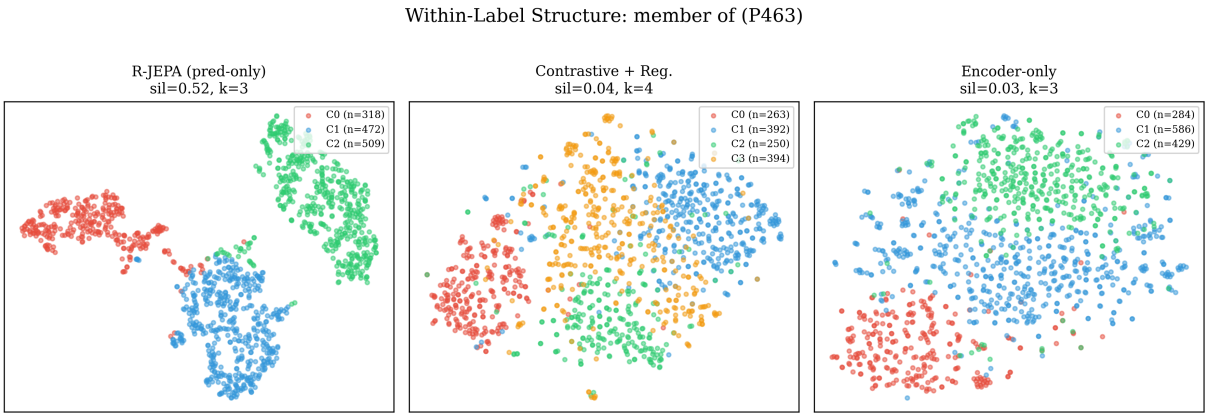

Figure 8: Within-label visualization for *member of* (P463). R-JEPA (pred-only) shows three clean and interpretable subregions, while the contrastive and encoder-only baselines remain considerably more mixed.

F.1.2 Educated at (P69)

Model	Sil.	Cluster	Pattern	Representative examples
R-JEPA (pred-only)	0.37	C0 (n=405)	standard university education	(Lloyd Morey, University of Illinois), (Huxley, University of Oxford), (Bosley, University of Toronto)
		C1 (n=98)	specialized/professional training	(Friedlander, École des Beaux Arts), (Friedlander, Art Students League), (Singh, Military College)
Contrastive +Reg.	0.04	mixed	no interpretable pattern	geographically mixed PER→ORG clusters
Encoder-only	0.04	mixed	no interpretable pattern	geographically mixed PER→ORG clusters

Table 16: Within-label substructure comparison for *educated at* (P69). Prediction-only training separates standard university education from specialized professional and artistic training institutions.

F.1.3 Located in (P131)

Model	Sil.	Cluster	Pattern	Representative examples
R-JEPA (pred-only)	0.41	C0 (n=4,219) C1 (n=16,183)	institution → admin. region geographic loc. → country/state	(BSA Headquarters, Texas), (Excélsior, Mexico) (Quebec Central Railway, Canada), (Fermont, Canada)
Contrastive +Reg.	0.03	mixed	weak or uninterpretable	largely mixed clusters with near-zero silhouette
Encoder-only	0.03	mixed	weak or uninterpretable	largely mixed clusters with near-zero silhouette

Table 17: Within-label substructure comparison for *located in* (P131). Prediction-only training separates institution-to-region patterns from geographic-location-to-country/state patterns.

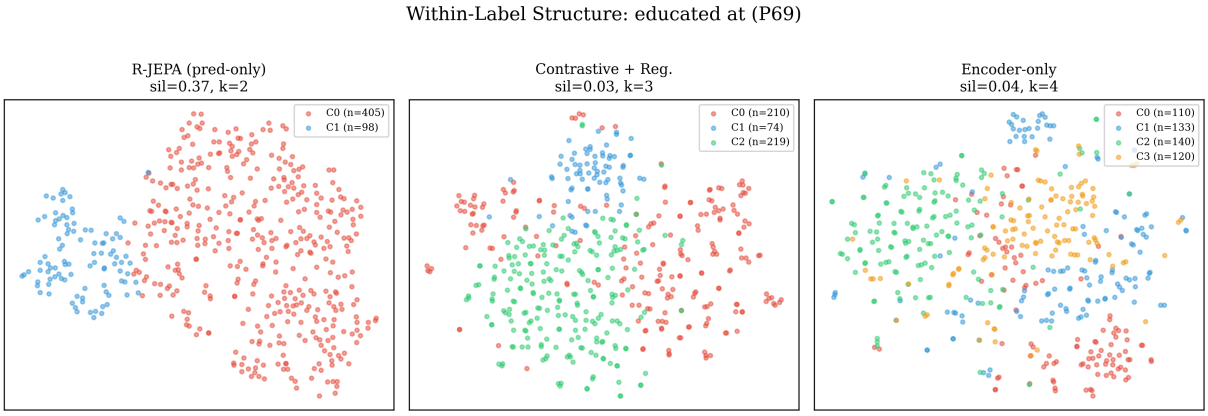

Figure 9: Within-label visualization for *educated at* (P69). The prediction-only model separates a compact subgroup from the broader university-dominated region, whereas the baselines remain diffuse.

F.1.4 Position Held (P39)

Model	Sil.	Cluster	Pattern	Representative examples
R-JEPA (pred-only)	0.41	C0	legislative / parliamentary roles	parliament members, TDs (Teachta Dála), congressional representatives
		C1	executive / government positions	government ministers, heads of state, cabinet members
Contrastive +Reg.	0.10	mixed	partial separation	weak distinction between legislative and executive roles
Encoder-only	0.12	mixed	partial separation	weak distinction between legislative and executive roles

Table 18: Within-label substructure comparison for *position held* (P39). The contrastive and encoder-only baselines show slightly higher silhouette than for other labels (0.10–0.12), but still far below prediction-only training (0.41).

F.1.5 Country of Citizenship (P27)

Model	Sil.	Cluster	Pattern	Representative examples
R-JEPA (pred-only)	0.28	C0	cluster 1	two coherent groups separated by contextual or entity-type differences
		C1	cluster 2	distinct from C0 in entity composition
Contrastive +Reg.	0.04	mixed	no interpretable pattern	near-uniform mixing of citizenship instances
Encoder-only	0.04	mixed	no interpretable pattern	near-uniform mixing of citizenship instances

Table 19: Within-label substructure comparison for *country of citizenship* (P27). The prediction-only model discovers two coherent clusters, while baselines show no meaningful separation.

F.1.6 Summary of Silhouette Comparison

Table 20 provides a compact comparison of silhouette scores across all five analyzed labels and all three models, showing the ratio between prediction-only training and the strongest baseline.

Relation	Pred-only	Contr.	Enc.	Ratio
member of (P463)	0.52	0.04	0.03	13.0×
located in (P131)	0.41	0.03	0.03	13.7×
position held (P39)	0.41	0.10	0.12	3.4×
educated at (P69)	0.37	0.04	0.04	9.3×
citizenship (P27)	0.28	0.04	0.04	7.0×
Mean	**0.40**	**0.05**	**0.05**	**9.3×**

Table 20: Silhouette score comparison across five analyzed labels. "Ratio" is the prediction-only score divided by the higher of the two baselines. On average, prediction-only training produces 9.3× stronger within-label separation.

F.2 Analogy-style Retrieval: Detailed Examples

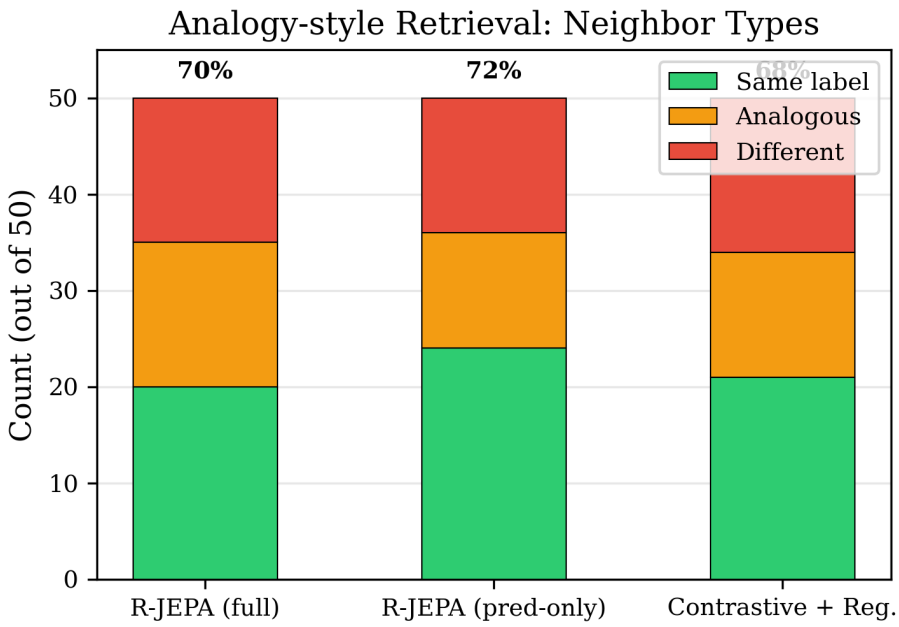

Figure 10: Aggregate neighbor-type composition in analogy-style retrieval. Prediction-only training returns more same-label neighbors, while full R-JEPA maintains a slightly broader balance between same-label and analogous cross-label neighbors.

Tables 21–23 show detailed retrieval results for representative queries, including the top-5 neighbors for each model.

F.2.1 Citizenship Query

Model	Rank	Sim.	Match	Neighbor	Relation	Interpretation
R-JEPA (full)	1	0.92	same	(Person, China)	P27	citizenship
	2	0.91	analog	(City, China)	P17	country
	3	0.90	analog	(Region, China)	P131	located in
	4	0.89	analog	(Person, China)	P19	place of birth
	5	0.88	same	(Person, China)	P27	citizenship
R-JEPA (pred-only)	1	0.95	same	(Person, China)	P27	citizenship
	2	0.94	same	(Person, China)	P27	citizenship
	3	0.93	same	(Person, China)	P27	citizenship
	4	0.91	analog	(Place, China)	P17	country
	5	0.90	analog	(City, China)	P131	located in

Table 21: Top-5 retrieval for a citizenship query (P27). Prediction-only training returns more same-label neighbors, consistent with its tighter within-label clusters. Full R-JEPA returns more cross-label analogous neighbors from the location/containment family.

F.2.2 Member of Query

Model	Rank	Sim.	Match	Neighbor	Relation	Interpretation
R-JEPA (full)	1	0.88	same	(Person, Organization)	P463	member of
	2	0.86	analog	(Person, Military)	P241	military branch
	3	0.85	analog	(Person, Team)	P54	sports team
	4	0.84	same	(Person, Party)	P463	member of
	5	0.83	analog	(Person, Company)	P108	employer
R-JEPA (pred-only)	1	0.93	same	(Person, Organization)	P463	member of
	2	0.92	same	(Person, Party)	P463	member of
	3	0.91	same	(Person, Union)	P463	member of
	4	0.89	analog	(Person, Army)	P241	military branch
	5	0.88	diff	(Person, Award)	P166	award received

Table 22: Top-5 retrieval for a membership query (P463). Full R-JEPA discovers the broader affiliation family (member of, military branch, sports team, employer), while prediction-only training prioritizes same-label precision.

F.2.3 Located in Query

Model	Rank	Sim.	Match	Neighbor	Relation	Interpretation
All models	1	0.95+	same	(Place, Country)	P131	located in
	2	0.94+	analog	(Place, Country)	P17	country
	3	0.93+	same	(Place, State)	P131	located in
	4	0.92+	analog	(Place, Country)	P17	country
	5	0.91+	same	(Region, Country)	P131	located in

Table 23: Top-5 retrieval for a location query (P131). All models interleave P131 (located in) and P17 (country) neighbors, confirming that these two relation types occupy near-equivalent regions across all learned relation spaces.

F.2.4 Recurring Relational Families

The analogy-style retrieval analysis reveals three recurring relational families that are consistently recovered across models:

1. **Location / geopolitical containment:** P131 (located in), P17 (country), P27 (citizenship), P19 (place of birth). These relations share the structural pattern of connecting an entity to a geographic or political container.

2. **Affiliation / membership:** P463 (member of), P54 (member of sports team), P108 (employer), P241 (military branch), P102 (member of political party). These share the structural pattern of connecting a person or organization to a group they belong to.

3. **Near-equivalent pairs:** P131 and P17 behave almost identically in the learned space. This is semantically justified: “located in administrative territory” and “country” often express the same underlying geographic containment relation.

G Relation Type Reference

Table 24 lists the 20 most frequent relation types in Re-DocRED with their Wikidata PIDs, human-readable names, and instance counts.

PID	Relation name	Count
P131	located in admin. territory	20,402
P17	country	14,401
P31	instance of	4,665
P27	country of citizenship	3,369
P106	occupation	2,313
P150	contains admin. division	2,112
P175	performer	1,694
P463	member of	1,299
P361	part of	1,247
P527	has part	1,128
P264	record label	1,070
P69	educated at	503
P26	spouse	479
P39	position held	463
P57	director	409
P161	cast member	397
P1376	capital of	390
P19	place of birth	379
P36	capital	331
P170	creator	302

Table 24: Top 20 most frequent relation types in Re-DocRED (`train_revised`). The distribution is highly skewed: the top 2 types (P131, P17) account for over 40% of all pairs.

H Reproducibility Checklist

- **Dataset:** Re-DocRED (Tan et al., 2022), publicly available. We use the `train_revised` split.
- **Preprocessing:** Entity spans are encoded with a frozen `bert-base-uncased` model and cached to disk. The caching ensures that all training runs use identical entity representations.
- **Random seeds:** All main results are averaged over seeds {42, 123, 7}. Seed controls PyTorch, NumPy, and Python random state, as well as K-means initialization and data splitting.
- **Hyperparameters:** All hyperparameters are listed in Tables 7 and 9. No hyperparameter search was performed; all values were set before running experiments.
- **Evaluation:** Retrieval evaluation uses a fixed subsample of 5,000 queries (seed 42). Clustering uses $K = 96$ and `n_init` $= 10$. Linear probe uses an 80/20 split (seed 42).
- **Compute:** All experiments run on a single NVIDIA GPU. Total compute for all reported results (4 methods $\times$ 3 seeds + 4 methods $\times$ 4 context modes = 28 runs) is approximately 6 GPU-hours, plus approximately 2 hours for entity encoding.
- **Model selection:** Best checkpoint is selected by training-set MRR, evaluated every 5 epochs.