A Graph Autoencoder Approach for Gesture Classification with Gesture AMR

Huma Jamil[1] **Ibrahim Khebour**[1] **Kenneth Lai**[2]
James Pustejovsky[2] **Nikhil Krishnaswamy**[1]
[1]Colorado State University, Fort Collins, CO USA
[2]Brandeis University, Waltham, MA USA
{huma.jamil,ibrahim.khebour,nkrishna}@colostate.edu
{klai12,jamesp}@brandeis.edu

Abstract

We present a novel graph autoencoder (GAE) architecture for classifying gestures using Gesture Abstract Meaning Representation (GAMR), a structured semantic annotation framework for gestures in collaborative tasks. We leverage the inherent graphical structure of GAMR by employing Graph Neural Networks (GNNs), specifically an Edge-aware Graph Attention Network (EdgeGAT), to learn embeddings of gesture semantic representations. Using the EGGNOG dataset, which captures diverse physical gesture forms expressing similar semantics, we evaluate our GAE on a multi-label classification task for gestural actions. Results indicate that our approach significantly outperforms naive baselines and is competitive with specialized Transformer-based models like AMRBART, despite using considerably fewer parameters and no pretraining. This work highlights the effectiveness of structured graphical representations in modeling multimodal semantics, offering a scalable and efficient approach to gesture interpretation in situated human-agent collaborative scenarios.

1 Introduction

In-person situated communication involves not just language, but non-verbal behavior like actions and, importantly, gestures. However, automated gesture interpretation is complicated by how the same gestural semantics may be represented by very different physical forms. Fig. 1 shows an instance of this: two people use entirely distinct iconic gesture shapes to denote the same concept—*block*.

This points to the need for higher levels of abstraction to adequately model the relationship between physical form and gestural meaning, particularly in collaborative dialogue. Abstract Meaning Representation (AMR; Banarescu et al. (2013)) is a popular choice in the computational semantics community for its clarity and expressiveness, and Brutti et al. (2022) and Donatelli et al. (2022) developed Gesture AMR (GAMR), an AMR formalism specifically for gesture semantics. Within GAMR, the semantics accompanying the iconic gesture *block*, irrespective of physical form, may be rendered as follows:

Figure 1: Example from the EGGNOG dataset (Wang et al., 2017) showing different gesture shapes expressing the same gesture semantics. Both are *iconic* gestures (Brutti et al., 2022) denoting blocks, articulated differently: the physical label of the left is `RH: into closed, left`; that of the right is `arms: move, up; hands: into facing, into open`.

```
(i / icon
    :ARG0 (s / signaler)
    :ARG1 (b / block)
    :ARG2 (a / actor))
```

In this paper, we observe that AMR/GAMR's natural graphical structure lends itself to graph neural network (GNN)-based approaches for automated processing, and propose a graph autoencoder (GAE) that learns mappings between gesture semantics represented in GAMR annotation and the physical forms of the associated gestures. Experiments on EGGNOG (Wang et al., 2017), a challenging audio-visual dataset, show that our approach both outperforms naive baselines, and beats or competes with strong Transformers on gesture shape prediction, despite having significantly fewer parameters and faster inference time, making our

Proceedings of the 16th International Conference on Computational Semantics, pages 41–48
September 22-23, 2025,

method suitable for gesture classification in low-resource and edge environments.

2 Related Work

Early work on gesture semantics followed traditions viewing gesture as simulated action (Kendon et al., 1980; Kendon, 2004) or a general mode of reference (McNeill, 1992, 2000, 2008). Following McNeill's work, Lascarides and Stone (2006, 2009) posited a division of gestures into *deictic* and *iconic*, creating a typing system continued in GAMR (Brutti et al., 2022; Donatelli et al., 2022). Lücking et al. (2015), Pustejovsky and Krishnaswamy (2021a,b, 2022), and Krishnaswamy and Pustejovsky (2021) further developed the grammar, semantics, and pragmatics of gesture on which GAMR is based. Related coding schemes for multimodal or non-verbal behavior include Kopp et al. (2006); Allwood et al. (2007); Kipp et al. (2007); Kong et al. (2015), and Rohrer et al. (2020).

Abstract Meaning Representation (AMR; (Banarescu et al., 2013)) is well-known for abstracting away from specific syntax using rooted, directed acyclic graphs (DAGs) and for applications to diverse tasks such as translation and NLU. Graph-based learning approaches using AMR include AMR-to-sequence learning (Beck et al., 2018) and text generation (Song et al., 2018; Wang et al., 2020; Zhao et al., 2020).

Our primary experimental dataset **EGGNOG** (Wang et al., 2017), containing natural gestures elicited during a collaborative task. EGGNOG has been used to train gesture recognizers for multimodal interactive agents such as Krishnaswamy et al. (2017, 2020, 2022) and Narayana et al. (2019). Lai et al. (2024) annotated a subset of EGGNOG with gesture and speech AMR, as well as coreference relations within and across the two modalities.

3 Methodology

The EGGNOG dataset (Wang et al., 2017) contains 360 videos of pairs of participants engaged in a collaborative task. One person, the actor, is given a set of wooden blocks, while the other, the signaler, is shown an image of a block structure. The signaler uses gesture, sometimes together with speech, to instruct the actor how to build the structure. Gestures are labeled according to both a *physical description* (e.g., `RH: thumbs, up`) and the signaler's *intent* (e.g., `yes`); this work focuses on the former.

Each EGGNOG physical gesture label refers to one or more body parts, which include the head, arms, hands, and upper body. Each body part is then described with one or more *aspects*, including various types of *motions* (of body parts in space, such as `rotate` and `shake`), *relations* (of body parts to each other, such as `crossed` and `facing`), and *poses* (hand positions, such as `claw` and `point`). Finally, aspects have optional orientations: `up`, `down`, `left`, `right`, `front`, or `back`. See Fig. 1 for an example. For simplicity, we focus on the *aspects* within each label.

Lai et al. (2024) annotated 21 of the EGGNOG videos with Gesture AMR. Because this was done separately from the physical gesture labels, a single GAMR can overlap with multiple labels. We link each GAMR with each overlapping label, and, in turn, with each aspect occurring in those labels, making this a multi-label classification problem. In total, the dataset contains 319 GAMRs (167 unique), associated with 889 aspects (33 unique). We split the data into an 80:20 train/test split.

3.1 Graph Autoencoder

Our graph autoencoder (GAE) learns graph-level representations from GAMR graphs for the EGGNOG classification task. It is adopted from the EdgeGAT-based message passing framework proposed by Zhang and Ji (2021), which leverages edge-aware attention mechanisms to integrate both node and edge features. Each node in the graph is represented using a one-hot 94D feature vector, where 94 is the size of the unique node vocabulary extracted from the GAMRs in the EGGNOG dataset. Edges are typed with one of 9 possible labels and are embedded into 9D continuous vectors using a learnable embedding layer. To enable bidirectional information flow between root and leaf nodes, all graphs are made explicitly bidirectional by adding the reverse of each original edge.

The encoder consists of three stacked EdgeGAT layers. Each EdgeGAT layer performs attention-based message passing where, for a given node i and neighbor j, attention score α_{ij} is computed as

$$\alpha_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T \left[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_j \,\|\, \mathbf{W}_e\mathbf{e}_{ij}\right]\right),$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are input node features, $\mathbf{e}_{ij}$ is the edge feature, and $\mathbf{W}$ and $\mathbf{W}_e$ are learnable linear projections applied to node and edge features, respectively. $\mathbf{a}^T$ is a learnable linear layer that maps the concatenated vector into a scalar attention score. Post-activation, these values are normalized using softmax to compute a weighted sum over neighbor

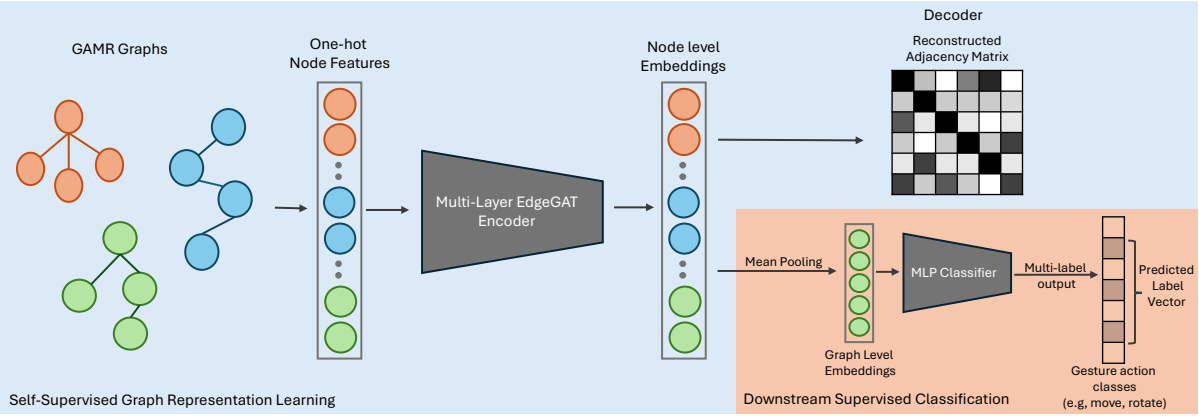

Figure 2: Graph autoencoder with EdgeGAT for self-supervised GAMR embedding, followed by MLP-based multi-label gesture classification.

embeddings. A residual connection is applied to preserve the original node features, controlled by a mixing parameter λ:

$$\mathbf{h}_i^{\text{out}} = (1-\lambda)\cdot\mathbf{h}_i + \lambda\cdot\sum_{j\in\mathcal{N}(i)}\text{softmax}_j(\alpha_{ij})\cdot\mathbf{W}\mathbf{h}_j.$$

Each EdgeGAT layer except for the last is followed by a ReLU activation. Node embeddings are then average-pooled into a fixed-dimensional graph representation $\mathbf{g} = \frac{1}{|V|}\sum_{i\in V}\mathbf{h}_i^{\text{final}}$, where V is the set of nodes and $\mathbf{h}_i^{\text{final}}$ is the embedding of node i from the last EdgeGAT layer.

We employ a multilayer perceptron (MLP) decoder to predict the presence of edges. For each edge (i, j), the decoder receives the concatenation of node embeddings $[\mathbf{z}_i \,\|\, \mathbf{z}_j]$ and outputs a scalar prediction $\hat{y}_{ij} = \sigma\left(\text{MLP}([\mathbf{z}_i \,\|\, \mathbf{z}_j])\right)$, where σ is the sigmoid activation function. The MLP consists of a 128D hidden layer, followed by ReLU, and a final linear layer projecting to a scalar.

The training objective is binary cross-entropy over observed positive and sampled negative edges:

$$\mathcal{L} = -\frac{1}{|E^+|}\sum_{(i,j)\in E^+}\log\hat{y}_{ij} - \frac{1}{|E^-|}\sum_{(i,j)\in E^-}\log(1-\hat{y}_{ij})$$

where E^+ denotes the set of observed edges and E^- is the set of randomly sampled negative edges. The model is optimized using the Adam optimizer with a learning rate of 0.001 over 100 epochs.

This GAE framework learns node and graph-level representations that capture both structural and semantic properties of the GAMR graphs. The learned graph embeddings are used for downstream classification in the EGGNOG task.

3.2 Evaluation

We evaluate the effectiveness of different vectorized GAMR representations for classifying the physical description of gestures. The EGGNOG dataset provides ELAN-annotated gesture instances along with their associated physical forms. High-level physical actions, such as *put*, *lift*, and *lean*, serve as the classification labels for this task.

The same GAMR (i.e., same graph structure) may appear multiple times across different gesture instances, each potentially annotated with a different set of physical labels. To investigate the impact of label aggregation on classification performance, we evaluate three label assignment strategies:

1. **Non-Aggregated (Instance-Level)**: Each GAMR instance is treated independently, with its own label set. This results in multiple instances of the same GAMR with potentially different labels.
2. **Majority Aggregation ($\geq$ 50%)**: For each unique GAMR, only those labels that appear in at least 50% of its instances are retained. This strategy aims to filter out noise while preserving consistent labels.
3. **Binary-Union Aggregation (Any Occurrence)**: For each unique GAMR, we include all labels that appear in **any** of its instances. This is the most inclusive strategy and ensures maximum label coverage.

All three versions result in a multi-label classification setup with 33 possible physical action aspect labels. We report results separately for each to enable informed choice of strategy for downstream task accuracy and robustness.

We compare classification performance of graph-based GAMR embeddings against several alternatives: (1) a naive baseline where GAMRs are represented using k-hot encodings of their node vocabulary, (2) embeddings of GAMRs extracted

	Instance-Level			Majority Aggregation			Binary-Union			
	P	R	F_1	P	R	F_1	P	R	F_1	# params.
k-hot	0.083	1.000	0.154	0.083	1.000	0.152	0.188	1.000	0.317	—
RoBERTa	0.475	0.479	0.477	0.602	0.599	0.601	0.772	0.833	0.802	124.1M
AMRBART	0.487	0.474	0.480	0.732	0.715	0.724	0.882	0.895	0.889	409.3M
GAE	0.490	0.447	0.468	0.731	0.648	0.687	0.834	0.836	0.835	52k

Table 1: Performance comparison of different GAMR representations across different label aggregation strategies on multi-label classification. All models use the same MLP classifier and training setup. *# params* incl. trainable and non-trainable, excl. MLP classification head.

from pretrained RoBERTa (Liu et al., 2019) using linearized AMRs as strings, and (3) GAMR embeddings from AMRBART (Bai et al., 2022) pretrained specifically on AMR parsing and generation.

For all embedding types, we use a lightweight multi-layer perceptron (MLP) classifier, consistent with common practice in unsupervised learning evaluations. The input to the classifier is the GAMR embedding vector as extracted from each method. All classifiers are trained and evaluated on the same 80:20 split described in Sec. 3.

All experiments follow the training protocol described in Sec. 3.1. This ensures that performance differences stem from the quality of the underlying GAMR representations rather than classifier capacity. We evaluate the three embedding types (GAE, AMRBART, RoBERTa), and the flat k-hot baseline, across the three aforementioned labeling strategies.

In these experiments, we use AMRBART-large-v2, which is a simpler, faster, and stronger version of AMRBART-large. This was pretrained on AMR 3.0[1], which comprises 55,635 training instances, as well as on 200,000 English sentences from English Gigaword[2]. RoBERTa experiments use RoBERTa-base.

4 Results and Discussion

Table 1 shows micro-averaged precision, recall, and F1 across all labels. The best overall performance is achieved under the **binary-union** label strategy, where a GAMR is labeled with any action that appears in at least one of its instances.

While AMRBART achieves the best F1 score overall, our GAE embeddings achieve competitive performance despite using orders of magnitude fewer parameters (Table 1, right side) and no pretraining. Notably, GAE embeddings outperform RoBERTa-based ones in both binary-union and majority aggregation settings, highlighting the benefit of incorporating relational structure over a linearized string representation. The naive k-hot baseline performs poorly all around due to its inability to encode structural context, and tends to overlabel all class, resulting in a spurious 100% recall. These results suggest that leveraging the graph structure of GAMRs provides a natural, effective and, notably, *efficient* way to learn meaningful gesture representations.

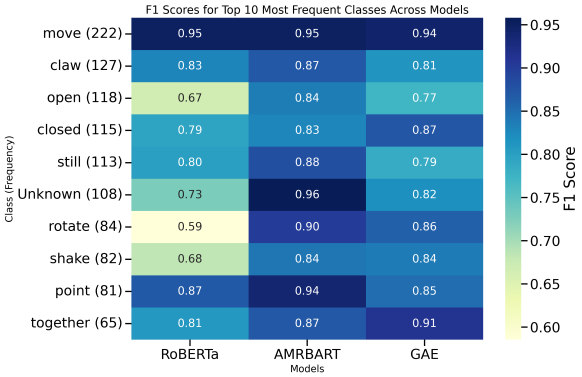

Figure 3: F1 scores for the top 10 most frequent classes across under binary-union labeling.

Table 2 shows the performance of our proposed method across the gesture types available from the EGGNOG dataset. We can observe a slight performance advantage leaning towards Iconic gestures when using instance-level labeling, which can be explained by the data imbalance toward this class as suggested by Table 3. However, under the binary-union strategy, Deixis gestures strongly outperform the other classes, this weakening the idea that the model might be biased towards any specific gesture category across labeling strategies. Instead, the strong performance of Deixis under this strategy may be attributable to the characteristic hand-shape of most deictic gestures that accompany English spoken dialogue.

Fig. 3 shows the F1 scores for the 10 most-frequently occurring physical gesture classes according to the binary-union strategy, across all

[1] https://catalog.ldc.upenn.edu/LDC2020T02

[2] https://catalog.ldc.upenn.edu/LDC2011T07

Gesture Type	Instance-Level			Majority Aggregation			Binary-Union		
	P	R	F_1	P	R	F_1	P	R	F_1
Iconic	0.525	0.489	0.506	0.692	0.537	0.605	0.624	0.653	0.638
Deixis	0.500	0.421	0.457	0.577	0.750	0.652	0.943	0.978	0.960
Emblem	0.450	0.474	0.462	0.348	0.727	0.471	0.524	0.942	0.674

Table 2: Performance comparison over different gesture types using the GAE method.

Gesture Types	Train	Test
Iconic	179	43
Deixis	54	14
Emblem	29	8

Table 3: Gesture types distribution across train and test sets.

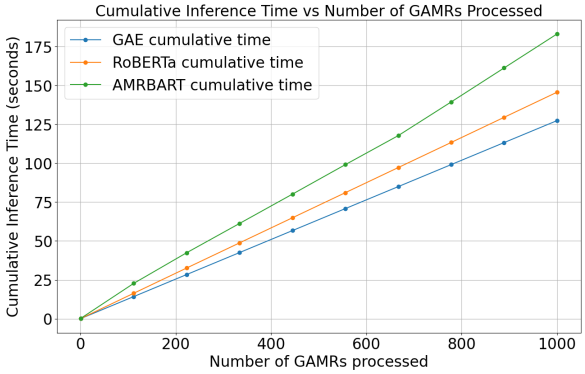

Figure 4: Cumulative inference time vs. number of GAMRs processed.

3 learnable methods. Here we see a number of classes where GAE embeddings match or exceed the performance of AMRBART embeddings, such as *closed*, *shake*, and *together*.

Finally, since the GAE has substantially fewer parameters than the competitor methods, we performed an inference-time experiment to quantify the speed advantage. Fig. 4 shows the cumulative time required to process increasing numbers of GAMRs by each method. We see that the GAE boasts a nearly 50% improvement in processing time over AMRBART despite AMRBART's extremely modest classification advantage, and that the GAE remains about 20% faster than RoBERTa at all input sizes despite outperforming it nearly globally.

5 Conclusion

We presented a novel approach to gesture classification using Gesture AMR and graph autoencoders. Our approach achieves competitive classification accuracy with SOTA Transformer approaches at significantly less computational overhead with faster inference speed. We also explored the effects of different label aggregation strategies, based on the premise that in real world data, the same semantics may have different physical forms attached to them. Our results can inform the choice of classification technique for downstream tasks that use gesture information with different requirements, such as epistemic position classification as in Khebour et al. (2024). Our efficient GAE method is suitable for real-time (e.g., VanderHoeven et al. (2025)) or GPU-less systems.

Limitations

Our method as presented (and all those tested) requires pre-annotated Gesture AMRs to be used as input, which entails additional human preparatory effort. Automating this step would entail some form of automatic AMR-graph construction for GAMR, such as sequence-to-graph transduction approaches for AMR parsing (Zhang et al., 2019) from raw dialogues and/or videos (VanderHoeven et al., 2024), potentially using text enrichment techniques such as dense paraphrasing (Tu et al., 2024).

Acknowledgments

This material is based in part upon work supported by Other Transaction award HR00112490377 from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program, and by award W911NF-25-1-0096 from the U.S. Army Research Office (ARO). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

Jens Allwood, Loredana Cerrato, Kristiina Jokinen, Costanza Navarretta, and Patrizia Paggio. 2007. The mumin coding scheme for the annotation of feedback, turn management and sequencing phenomena. *Language Resources and Evaluation*, 41:273–287.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for amr parsing and generation.

In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Lucia Donatelli, Kenneth Lai, Richard Brutti, and James Pustejovsky. 2022. Towards situated amr: Creating a corpus of gesture amr. In *International Conference on Human-Computer Interaction*, pages 293–312. Springer.

Adam Kendon. 2004. *Gesture: Visible action as utterance*. Cambridge University Press.

Adam Kendon et al. 1980. Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25(1980):207–227.

Ibrahim Khebour, Kenneth Lai, Mariah Bradford, Yifan Zhu, Richard Brutti, Christopher Tam, Jingxuan Tu, Benjamin Ibarra, Nathaniel Blanchard, Nikhil Krishnaswamy, and James Pustejovsky. 2024. Common ground tracking in multimodal dialogue. *arXiv preprint arXiv:2403.17284*.

Michael Kipp, Michael Neff, and Irene Albrecht. 2007. An annotation scheme for conversational gestures: how to economically capture timing and form. *Language Resources and Evaluation*, 41:325–339.

Anthony Pak-Hin Kong, Sam-Po Law, Connie Ching-Yin Kwan, Christy Lai, and Vivian Lam. 2015. A coding system with independent annotations of gesture forms and functions during verbal communication: Development of a database of speech and gesture (dosage). *Journal of nonverbal behavior*, 39:93–111.

Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R Thórisson, and Hannes Vilhjálmsson. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents: 6th International Conference, IVA 2006, Marina Del Rey, CA, USA, August 21-23, 2006. Proceedings 6*, pages 205–217. Springer.

Nikhil Krishnaswamy, Pradyumna Narayana, Rahul Bangar, Kyeongmin Rim, Dhruva Patil, David McNeely-White, Jaime Ruiz, Bruce Draper, Ross Beveridge, and James Pustejovsky. 2020. Diana's world: A situated multimodal interactive agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13618–13619.

Nikhil Krishnaswamy, Pradyumna Narayana, Isaac Wang, Kyeongmin Rim, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Ross Beveridge, Jaime Ruiz, Bruce Draper, et al. 2017. Communicating and acting: Understanding gesture in simulation semantics. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)—Short papers*.

Nikhil Krishnaswamy, William Pickard, Brittany Cates, Nathaniel Blanchard, and James Pustejovsky. 2022. The voxworld platform for multimodal embodied agents. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1529–1541.

Nikhil Krishnaswamy and James Pustejovsky. 2021. The role of embodiment and simulation in evaluating hci: Experiments and evaluation. In *International Conference on Human-Computer Interaction*, pages 220–232. Springer.

Kenneth Lai, Richard Brutti, Lucia Donatelli, and James Pustejovsky. 2024. Encoding gesture in multimodal dialogue: Creating a corpus of multimodal AMR. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5806–5818, Torino, Italia. ELRA and ICCL.

Alex Lascarides and Matthew Stone. 2006. *Formal semantics for iconic gesture*. Universität Potsdam.

Alex Lascarides and Matthew Stone. 2009. A formal semantic analysis of gesture. *Journal of Semantics*, 26(4):393–449.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andy Lücking, Thies Pfeiffer, and Hannes Rieser. 2015. Pointing and reference reconsidered. *Journal of Pragmatics*, 77:56–79.

David McNeill. 1992. Hand and mind. *Advances in Visual Semiotics*, 351.

David McNeill. 2000. *Language and gesture*, volume 2. Cambridge University Press.

David McNeill. 2008. Gesture and thought.

Pradyumna Narayana, Nikhil Krishnaswamy, Isaac Wang, Rahul Bangar, Dhruva Patil, Gururaj Mulay, Kyeongmin Rim, Ross Beveridge, Jaime Ruiz, James Pustejovsky, et al. 2019. Cooperating with avatars through gesture, language and action. In *Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 1*, pages 272–293. Springer.

James Pustejovsky and Nikhil Krishnaswamy. 2021a. Embodied human computer interaction. *KI-Künstliche Intelligenz*, 35(3):307–327.

James Pustejovsky and Nikhil Krishnaswamy. 2021b. The role of embodiment and simulation in evaluating hci: theory and framework. In *International Conference on Human-Computer Interaction*, pages 288–303. Springer.

James Pustejovsky and Nikhil Krishnaswamy. 2022. Multimodal semantics for affordances and actions. In *International Conference on Human-Computer Interaction*, pages 137–160. Springer.

Patrick Louis Rohrer, Ingrid Vilà-Giménez, Júlia Florit-Pons, Núria Esteve-Gibert, Ada Ren, Stefanie Shattuck-Hufnagel, and Pilar Prieto. 2020. The multimodal multidimensional (m3d) labelling scheme for the annotation of audiovisual corpora. *Gesture and Speech in Interaction (GESPIN)*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Jingxuan Tu, Kyeongmin Rim, Bingyang Ye, Kenneth Lai, and James Pustejovsky. 2024. Dense paraphrasing for multimodal dialogue interpretation. *Frontiers in artificial intelligence*, 7:1479905.

Hannah VanderHoeven, Brady Bhalla, Ibrahim Khebour, Austin C Youngren, Videep Venkatesha, Mariah Bradford, Jack Fitzgerald, Carlos Mabrey, Jingxuan Tu, Yifan Zhu, et al. 2025. Trace: Real-time multimodal common ground tracking in situated collaborative dialogues. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 40–50.

Hannah VanderHoeven, Nathaniel Blanchard, and Nikhil Krishnaswamy. 2024. Point target detection for multimodal communication. In *International Conference on Human-Computer Interaction*, pages 356–373. Springer.

Isaac Wang, Mohtadi Ben Fraj, Pradyumna Narayana, Dhruva Patil, Gururaj Mulay, Rahul Bangar, J Ross Beveridge, Bruce A Draper, and Jaime Ruiz. 2017. Eggnog: A continuous, multi-modal data set of naturally occurring gestures with ground truth labels. In *2017 12th Ieee international conference on automatic face & gesture recognition (fg 2017)*, pages 414–421. IEEE.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020. AMR-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. Amr parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 732–741, Online. Association for Computational Linguistics.

A Additional Technical and Implementation Details

Hyperparameters Each of the 3 layers of the EdgeGAT network consists of 94 hidden units. The value of the mixing parameter λ was set to 0.5. The MLP classifier consists of 3 hidden layers (256, 128, and 64 units, respectively). Batch normalization and ReLU activation are used after each of the first three linear layers, followed by a dropout layer with probability 0.3. The activation function used throughout is ReLU.

MLP Decoder An inner product decoder models only a simple, fixed linear similarity between node embeddings. That is, it only predicts that an edge exists between two nodes if node vectors are aligned (high inner product), providing a rigid notion of connectivity. By contrast, an MLP provides a learnable decoder which can learn complex, nonlinear relationships to explain the presence and absence of edges, and hence can more reliably capture asymmetric relationships. When comparing inner product with MLP approaches during development, we used AUROC on the task of reconstructing the node adjacency matrix as a guiding metric. An inner product decoder achieved a top AUROC of 93, which increased to 99 with the MLP decoder.

Sampling Strategy Random sampling was used for sampling negative edges for training (Sec. 3.1). For each batch, we sampled node pairs that are not connected in the input graph to serve as negative edges. The sampling is uniform and done on the fly during training and evaluation. We use the `negative_sampling` utility by `torch_geometric`, which makes sure that sampled edges do not overlap with positive edges.

Hardware and Software All classification experiments were performed on an AMD Ryzen Threadripper 3960X 3.8 GHz system with 96 GB RAM running Linux 5.15.0-130-generic x86_64 (Ubuntu-based kernel).

The inference time experiment shown in Fig. 4 was performed on an Intel Xeon Gold 5520+ with 256 GB RAM and Ubuntu 24.04.2 LTS.

PyTorch 2.4.0 was used.