

MemeArgs: Vision-Language Argument Mining in Persuasive Memes

Khoi P. N. Nguyen, Trung N. Nguyen, Terrence Li, Teerth Patel, Vincent Ng

Human Language Technology Research Institute

University of Texas at Dallas

khoi.nguyen6@utdallas.edu, vince@hlt.utdallas.edu

Abstract

On today’s social media, many argumentative artifacts appear in mixed image-text format. Meanwhile, the argument mining community has seen much success in developing argument understanding technologies in text. In this work, we bridge that gap by studying vision-language argument (VLA) parsing: generating an argument graph that captures both the explicit and implicit components of a vision-language argument. To lay the groundwork for this task, we annotate MemeArgs¹, a corpus of argument graphs for 614 persuasive memes. We then propose an automatic evaluation protocol for VLA parsing that accommodates its generative nature and follows standard argument mining practice. Evaluating current general-purpose vision-language models, we find that they all fail to perform knowledge-grounded visual perception reliably, which distorts their understanding of the arguments. VLA parsing thus presents an exciting challenge for multi-modal argument understanding.

1 Introduction

Argument mining (AM) is the task of automatically identifying and representing the structure of arguments in text (Liu et al., 2023a). By mapping out claims, premises, and their relationships, AM enables a deeper understanding of how ideas are constructed, supported, and contested in discourse.

Beyond natural language, arguments nowadays tend to exist in multimodal forms, typically as a combination of image and text. Opinionated memes, visual propaganda, and photographic advertisements are prominent categories of media where arguments are advanced via image and text. In the U.S., for instance, political manipulation campaigns aiming to sway public opinion have been prevalent before major elections (Bradshaw

¹Our data and code are publicly available at <https://github.com/npnkhoi/memeargs>

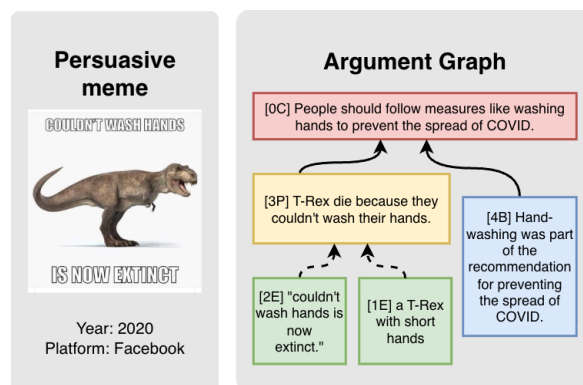


Figure 1: **A Vision-Language Argument Graph.** The input is a persuasive meme. The resulting graph represents the argument structure of the original image. Color meanings: green box=element, blue box=background, yellow box=premise, red box=claim, solid arrow=support edge, dashed arrow=trigger edge.

and Howard, 2018). Part of these efforts involves disseminating disinformation via *visio-linguistic artifacts*, such as opinionated memes and visual propaganda, designed to unconsciously manipulate their viewers on social media platforms. Automatic analysis of the structure of the arguments made in such artifacts could help the public understand not only what the malicious intent is but also how they are misled (Ng and Li, 2023). The same analysis has value in advertising: making an ad’s argument structure explicit helps consumers see past emotionally charged messaging, and lets advertisers check whether they are communicating the merits of a product rather than manipulating their audience.

Given the prevalence of visio-linguistic arguments and their potential for manipulation, it is natural to extend AM technologies to process them. We call this problem class *Vision-Language Argument Mining* (henceforth, VLAM). In this work, we tackle *Vision-Language Argument Parsing* (VLA parsing), which we define as the task of automatically discovering the argumentative structure of a persuasive text-augmented image (i.e., an im-

age in which a possibly hidden argument is made). VLA parsing is the core problem of VLAM. We target the *end-to-end* setting, in which a model maps an image directly to its vision-language argument graph (henceforth, VLA graph). To illustrate, consider Figure 1. Given the persuasive image shown on the left (in this case, a meme), the goal of VLA parsing is to produce the VLA graph shown on the right. It lays out the argument structure of the original image, including the claims and the evidence, as well as the relationships between them. We will formally define VLA graphs in Section 3.

VLA parsing takes AM from its traditional text-based setting to the next level of complexity. In text-based AM, the argument components (i.e., the claims and premises) are typically expressed explicitly in the associated text and therefore can be largely *extracted*. In contrast, multimodal arguments are often implied by the visual elements, the text-image interactions, and the context around them. Consequently, most, if not all, of the argument components, particularly those that correspond to the non-leaf nodes of an argument graph, need to be *generated* rather than *extracted*.

In this paper, we make several contributions to the foundation of VLAM. First, we propose end-to-end VLA parsing, the task of generating an argument graph from a persuasive image. Second, we present MemeArgs, the first corpus for VLA parsing composed of 614 persuasive memes. Third, we propose an automatic protocol for evaluating generated VLA graphs and show that it correlates well with human judgments. Finally, we present the first set of results on end-to-end VLA parsing for persuasive memes using a range of state-of-the-art vision-language models.

2 Related Work

Text-based argument mining There has been a long tradition of work in text-based AM. Central to this field is the automatic construction of argument graphs, which typically comprises four steps: text segmentation, argument/non-argument classification, simple structure construction, and structure refinement. Early approaches to AM were based on hand-crafted features (Moens et al., 2007; Nguyen and Litman, 2016) or custom neural architectures (Stab and Gurevych, 2017; Liu et al., 2023a). Meanwhile, recent approaches make use of pretrained transformer models (Kawarada et al., 2024; Favero et al., 2025; Cabessa et al., 2025).

To facilitate the development of AM systems, corpora of argument graphs have been created on different types of text, such as legal text (Reed, 2005), student essays (Stab and Gurevych, 2014), microtexts (Peldszus and Stede, 2015), science papers (Kirschner et al., 2015), and financial comments (Liu et al., 2024). A detailed overview of text-based AM research can be found in Cabrio and Villata (2018) and Lawrence and Reed (2019).

Visual argument theories In the 1990s, argumentation theorists started to pay attention to visual argumentation. While there is a common consensus that visual artifacts "can play an important role in argumentation", much debate persists on a working theory for visual arguments (Kjeldsen, 2015). The prominent schools of thought are Toulmin's (2003) model, van Eemeren and Grootendorst's (2016) pragma-dialectics, and Johnson and Blair's (1977) approach. Among them, the Toulmin model is seeing growing adoption, being used to represent arguments in political ads (Kjeldsen, 2007), political cartoons (Groarke, 2009), and commercial ads (Ripley, 2008). When this model is applied to a persuasive image, the premises, though visual, can be textualized into argument components, in particular as Backing/Warrant and Ground components. Nevertheless, some limitations of the Toulmin model exist when it is applied to visual arguments. Ripley (2008) pointed out that analyzing an emotional, visceral, or kisceral argument using popular argumentation models (including Toulmin's) causes the argument to lose some of its power. Meanwhile, Groarke (2009) noted that the Toulmin model might fail to capture humor, which is non-logical but is a key to the rhetoric of political cartoons.

Multimodal argument mining Researchers have recently begun work on discovering arguments in vision-language (VL) data, but have thus far focused on VLAM-related tasks only. For instance, the ImageArg dataset was introduced to evaluate the persuasiveness of the arguments made in image-accompanied tweets (Liu et al., 2022). This dataset was extended to become the first shared task on Multimodal Argument Mining (Liu et al., 2023b) at the 10th Workshop on Argument Mining, which focused on two key tasks: (1) classifying the stance implied by the argument made in a given image and (2) determining argument persuasiveness. The winning system employed a custom architecture that encodes text, images, and lay-

outs (Zong et al., 2023). Beyond vision-language, Mestre et al. (2021) proposed audio-textual argument understanding and released data from the 2020 US presidential debates.

Reasoning with images To facilitate automatic reasoning with cartoons, Chung et al. (2024) provided a dataset where cartoons were annotated with reasoning graphs. These reasoning graphs show the reasoning steps involved in deriving the *intent* expressed by the cartoon’s author. Though reasoning graphs and argument graphs bear a resemblance, they are fundamentally different conceptual structures. While the non-leaf nodes of an argument graph correspond to claims, those of a reasoning graph may be non-argumentative statements. For instance, the reasoning graph constructed from the meme in Figure 1 would consist of a root that corresponds to "The author uses a humorous reason to highlight the necessity of washing hands during COVID-19," which is the author’s intent and does not directly express an argument. Despite their annotations, Chung et al. did not propose or evaluate any method for constructing reasoning graphs.

3 Vision-Language Argument Graph

In this section, we formally define the output of the VLA parsing task: the VLA graphs.

3.1 Definition

A VLA graph is a directed acyclic graph that reveals the argument structure underlying a text-overlaid persuasive image (Figure 2).

Each **node** of a VLA graph contains text that corresponds to a component of the overall argument made in the image and can be categorized as one of the following: (1) a **literal element** (or *element*), which describes some observable aspect of the image (i.e., textual and visual elements); (2) a **premise**, which states a proposition triggered by the literal elements; (3) a **claim**, which makes some assertion that is justified or challenged by at least one other node; or (4) **background knowledge and belief** (or *background, BK*), which describes information that is *not* present in the image, but serves as a premise needed to eventually establish the overall claim made in the image.

Each (directed) **edge** of a VLA graph represents an ordered relationship between two nodes and can be categorized into one of three types: (1) a **support** edge represents how a claim is supported by a premise, background, or an already established

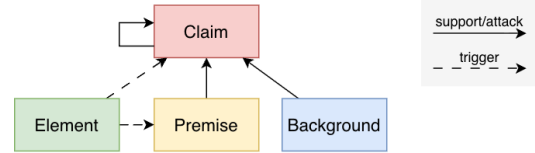


Figure 2: **The scheme of VLA graphs.** Solid lines are support/attack edges. Dashed lines are trigger edges.

claim in the graph; (2) an **attack** edge represents a rebuttal to a previous claim; and (3) a **trigger** edge connects an element node to other claims or premises that it evokes.

Note that element nodes, which describe the textual and visual elements in the image, should not be seen as propositions as in verbal arguments. For example, an element with the content "John is smart" is referring to the region of the image containing that sentence, not the idea that John is smart. From this perspective, elements do not "support" or "attack" claims: they are purely visual details that trigger the premises and claims via trigger edges, which carry emotions, sarcasm, and other rhetorical techniques often seen in visual arguments.

To complete this scheme, we impose three well-formedness constraints. First, there must be exactly one claim node with an out-degree of zero, which corresponds to the conclusion of the overall argument. Second, the graph must be weakly connected, meaning that there must be an undirected path between any two nodes in the graph. This is because all the nodes in the graph are part of the same argumentative structure and should be connected to each other in some way. Third, a trigger edge must go from an element to a claim or premise.²

Our schema is largely consistent with the classical argumentation graph scheme used in Stab and Gurevych (2014) for annotating persuasive essays and extended by Habernal et al. (2018), itself an instantiation of the Toulmin model (Toulmin, 2003). In their scheme, the *major claim* is the main point of the essay, supported by some *arguments*. Each argument contains a *claim*, which is supported or attacked by other *premises*. Note that, in their scheme, a premise can be supported or attacked by other premises. In our scheme, we call those higher-level premises intermediate claims to make it clear that they are logically supported or attacked by other nodes (and possibly triggered by the elements). Finally, their *warrants* are implicit facts or beliefs that correspond to our background nodes.

²Implementation-wise, the VLA graph can be represented in a JSON schema (see Appendix A).

3.2 Example

Consider the VLA graph in Figure 1 (right). Going from the bottom, we can see that the graph cites two surface visual **elements** of the image: ‘[1E] a T-Rex with short hands’ and ‘[2E] “couldn’t wash hands is now extinct”’. These elements trigger the meme’s **premise** that ‘[3C] T-Rexes died because they couldn’t wash their hands’.

From here, given that the image was posted on Facebook in 2020, we know that it assumes the audience knows the **background knowledge** that ‘[4B] Hand-washing was part of the recommendation for preventing the spread of COVID.’ This shared context is crucial for proper interpretation of the image. Both 3C and 4B then come together to support the major **claim** that ‘[0C] People should follow measures like washing hands to prevent the spread of COVID.’

4 The MemeArgs Corpus

4.1 Memes

Among the different kinds of visio-linguistic artifacts, we chose to annotate *persuasive memes* with VLA graphs. Memes are user-created combinations of images overlaid with text, which have become a prevalent means of online communication (Milner, 2018). Our decision to annotate memes is based in part on their societal importance: while some memes are simply used to express personal opinions, other memes can be malicious, such as inciting hatred or spreading manipulative propaganda. As such, recent years have seen increasing interest within NLP in their automatic processing.

To assemble our dataset, MemeArgs, we sampled 640 memes from the SemEval2021 Task 6 (Dimitrov et al., 2021) corpus, which contains political memes from the US and Canada collected on Facebook during 2020. They contain English text and cover a variety of topics, including politics, conspiracy theories and misinformation, health and science skepticism, social and cultural issues, public health, media criticism and information control, and foreign policy.

4.2 Annotation Procedure

To construct argument graphs for the memes, we designed a five-stage annotation procedure. For each stage, we collected *two* independent annotations per meme, and had the first and second authors act as experts to resolve disagreements and ensure

quality before moving on to the next stage.³

Stage 1: Annotating the conclusion (c^*) For each meme, two annotators independently wrote a statement that captures the conclusion (i.e., major claim) of the meme’s argument. To contextualize the meme, we provided the year and the platform where the meme was posted, and instructed the annotators to think from the perspective of someone who posted the meme in such a context.⁴

An expert (the first author) evaluated the agreement on the conclusion annotations. For each meme, they rated whether the two claims were semantically equivalent (i.e., they conveyed the same meaning and can be used interchangeably in the context of the meme) or not. For the disagreeing pairs, the expert decided to either keep one of the two or rewrite the conclusion completely. Twenty-six memes were deemed to be genuinely ambiguous, i.e., both conclusions are plausible without further context. We discarded these instances and were left with 614 instances, each with a conclusion c^* .

Stage 2: Annotating elements (E) Next, we annotated the visually salient elements in the meme needed to establish the conclusion. Annotators were instructed to write down all *unique* and *relevant* objects, people, their qualities/actions, or text fragments, each into one element node.

During review, for each meme, each expert was given two sets of nodes annotated by two annotators, which we call A and B . They conducted a pairwise comparison of the nodes in A and B to determine which nodes were semantically equivalent, using the same criteria as in stage 1. A 1-to-1 mapping was then established between the nodes in A and B based on these equivalence judgments. For nodes that were not matched in each meme, annotators discussed whether they should be merged with another node (due to semantic equivalence), kept separate (unique and relevant), or discarded (irrelevant or incorrect). Nodes matched earlier were automatically merged. After this process, each meme has a set of element nodes $E = \{e_i\}$ that were semantically distinct and agreed upon by the annotators.

³See Appendix B about annotator recruitment and training and Appendix C for the annotation guidelines.

⁴By providing this contextual information, we aimed to guide the annotators in interpreting the meme’s argument in a way that is faithful to its original intent and the socio-political environment in which it was created.

Stage 3: Annotating premises and intermediate claims (C) Next, we collected premises and intermediate claims, i.e., internal nodes in the graphs that connect the leaves with the root. To do so, we tasked each pair of annotators with independently constructing a graph in a bottom-up fashion, starting from the leaf nodes (the E nodes collected in stage 2) and building up towards the root node (i.e., the c^* node collected in stage 1).

During review, the experts treated the new premises and claims as a set of new nodes, ignoring the attached edges. The idea is to obtain a unified list of distinct and relevant new nodes. As such, the experts applied the same protocol as in stage 2, working with the annotators to remove invalid, irrelevant, and duplicated new nodes, ending up with a single set of new nodes $C = \{c_j\}$.

Stage 4: Annotating Edges ($\hat{G}$) For each meme, given its c^* , E , and C , the annotators connected the nodes with support, attack, and trigger edges. While it seems like we are asking the annotators to re-do what they did in stage 3, we believe this stage is necessary since it is difficult to resolve disagreements on edges without having a set of nodes the annotators agreed upon. We then asked the annotators to resolve the disagreements. At the end, each meme has a preliminary graph $\hat{G}$.

Stage 5: Annotating background knowledge (G) Given $\hat{G}$, the annotators were asked to identify for each argument all relevant pieces of background knowledge (BK), where an argument is composed of a claim and all of its children. A BK node must be added if there are hidden presumptions (background knowledge) that the audience was assumed to have in order to establish the claim in the argument. The challenge here lies in defining when a piece of BK is relevant enough to be added. Motivated by Habernal et al. (2018), who annotated "warrants" (which correspond to BK) for text-based AM, we utilized their "alternative warrant" trick, where a piece of background knowledge is relevant if, in a world where the opposite were true (i.e., alternative warrant), the claim would be wrong.

Disagreement was resolved at each argument step where BK was added by at least one annotator. The protocol is similar to that of stages 2 and 3, where the expert provided the matching and the annotators resolved the disagreements.

4.3 Inter-Annotator Agreement

At the end of every annotation stage, we measured inter-annotator agreement (IAA) to assess the reli-

Node type	ΣN	$\#N/G$	ΣW	$\#W/N$
Element	1701	2.77	20.3K	11.95
Premise	131	0.21	1.5K	11.47
Claim	1432	2.33	15.9K	11.13
Background	1537	2.50	27.5K	17.87
Total	4801	7.82	65.2K	13.59

Table 1: **Statistics per node type in MemeArgs.** ΣN , $\#N/G$, ΣW , and $\#W/N$ mean total number of nodes, average number of nodes per graph, total number of words, and average number of words per node, respectively.

ability of our annotation procedure.⁵ These were computed before merging the annotations.

In stage 1, of 640 original pairs of conclusions, 573 were rated as semantically equivalent by the two experts, resulting in an agreement rate of **89.5%**. This indicates high agreement. For stages 2, 3, and 5, we measured agreement on the annotated node sets using F1 scores.⁶ The macro-average F1 scores across the 614 memes were **94.9%** for elements, **66.4%** for premises and intermediate claims, and **52.9%** for BK nodes, which implies high agreement on visually grounded elements and moderate agreement on higher-level argumentative components. As identifying claims and background information requires deeper reasoning and more subjective judgments, their IAA values are understandably lower than those of the elements. All those disagreements have been resolved by the experts.

To compute IAA on edges, we followed the common practice in AM and compute Cohen’s κ , treating all pairs of nodes in the given node set as markables (Stab and Gurevych, 2014; Lawrence and Reed, 2019). Two annotations agree if they both assigned an edge with the same direction and type, or both not assigning an edge. We obtained $\kappa = 0.89$, indicating substantial agreement.

4.4 Final Dataset

The final corpus contains 614 graphs, spanning over 4,801 nodes and 4,749 edges, resulting in over 65.2K words. The graph statistics further show the sophistication of the argumentative structures in our data. Each graph on average has 7.82 nodes. Non-element nodes, i.e., those that are not literally presented in the memes, on average take up 64.7% of the nodes in each graph. Table 1 shows statistics

⁵See Appendix D for sources of annotator disagreement.

⁶See Appendix E for details on how F1 scores were computed.

Edge type	ΣE	#E/G
Support	2636	4.29
Attack	17	0.03
Trigger	2096	3.41
Total	4749	7.73

Table 2: **Statistics per edge type in MemeArgs.** ΣE and #E/G are the total number of edges and the average per graph.

Corpus	Genre	#Doc	#Comp
Stab and Gurevych (2014)	student essays	90	1552
Peldszus and Stede (2015)	microtexts	112	576
Kirschner et al. (2015)	science papers	24	2700
Stab and Gurevych (2017)	student essays	401	6089
MemeArgs	memes	614	4801

Table 3: **Comparison of MemeArgs with four major text-based argument graph corpora for English.** (Abbreviations: #Doc: number of documents (or memes), #Comp: number of argument components)

for each node type. Table 2 breaks the 4,749 edges down by type. Support and trigger edges are the bulk of the corpus, at 4.3 and 3.4 per graph; attack edges—propositions rebutting a claim—are rare, appearing in only 14 graphs.

The size of MemeArgs is typical for settings where each instance requires careful, fine-grained human annotations and is sufficient to support statistically meaningful comparisons. Table 3 compares MemeArgs with four well-known text-based AM corpora for English. Among them, MemeArgs is the largest in terms of the number of documents. Keep in mind that it is more difficult to annotate VLA graphs than text-based ones because claims need to be inferred rather than extracted.

In terms of topics, the conclusions in the dataset belong to two major groups: (1) **US/Canadian politics** — Trump, Obama, Democrats, Republicans, individual politicians (Pelosi, McConnell, Trudeau, etc.); and (2) **Science and pseudoscience** — vaccines/anti-vaxxers, alternative medicine (homeopathy, naturopathy, acupuncture, reiki, cupping, aromatherapy), GMOs, climate change denial, flat earth/moon landing conspiracies, 5G, and nutrition myths (glyphosate, vitamin megadosing, natural food labels). Outside those two, a small number of memes cover a handful of isolated topics: domestic violence, depression, religion, and the environment. Nearly every conclusion is an evaluative or normative claim supporting or attacking a person, group, or belief system.

We created a train-test split for MemeArgs

roughly at a 60:40 ratio as follows. For each graph, we first linearized its nodes in topological order and obtained their embeddings using the BGE-M3 model (Chen et al., 2025). We then performed single-link agglomerative clustering on the graphs, using cosine similarity as the similarity function. During clustering, the first cluster that reached 40% of the total dataset size was designated as the test set, with the remaining 60% forming the training set. This approach ensures that memes with similar meanings are kept together, reducing the risk of semantic overlap between the training and test sets.⁷ The final training and test splits have 348 and 266 instances, respectively.

5 Evaluation

We evaluated on MemeArgs a range of leading vision language models to gauge the difficulty of VLA parsing.

5.1 Performance Measures

Node Metrics First, we computed the rate at which the generated conclusion matched the ground truth, which we call the **Conclusion Matching Rate**. Next, motivated by the standard protocol used in evaluating text-based AM systems (Persing and Ng, 2016), we computed the node F1 score for each of the four node types, yielding **Element F1**, **Background F1**, **Premise F1**, and **Claim F1**. Specifically, given a meme in the test set, we first computed the Node F1 score for each node type from the corresponding gold (i.e., human-annotated) graph and the model-generated graph, and then macro-average per-type node F1 scores over the test memes to obtain the final node F1 scores.⁸

Computing these scores requires the nodes in the two graphs to be matched. In text-based AM, node matching is computed based on *token overlap* in the corresponding text spans, both of which are extracted from the underlying text. In VLA parsing, because nodes are largely generated, we must rely on *semantic equivalence*. At a high level, two nodes are semantically equivalent if they have the same node type and discuss the same entities with the same quality or action. We perform node semantic matching using Gemini 3 Flash (Gemma Team, 2025) with the prompt in Appendix F.

⁷The maximum similarity between a training set instance and a test set instance is 0.6634.

⁸See Appendix E for details on how the node F1 scores were computed.

Model	Concl.	Element			Premise			Claim			Background			Logos edge			Trigger edge			GED
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
InternVL2.5-26B	13.2	61.4	67.9	62.1	55.6	78.9	41.0	26.9	23.9	23.0	28.8	21.2	16.6	6.9	4.0	3.3	19.2	13.4	14.0	21.7
Qwen2.5-VL-32B	39.8	77.7	76.3	74.1	66.5	82.9	55.3	46.9	40.6	40.3	40.0	37.0	31.9	16.4	13.2	12.5	36.1	28.0	30.0	17.7
Mistral-Small-24B	48.6	75.9	86.7	78.4	86.4	79.4	68.1	48.4	44.9	43.4	39.4	44.6	37.0	15.4	15.2	13.9	34.0	27.5	29.5	17.1
Gemma-3-27B	56.8	78.1	84.3	78.8	71.8	83.1	59.9	53.6	48.2	47.5	53.4	46.4	43.6	23.2	20.1	20.0	39.8	33.3	35.0	15.4
Grok-4.20	60.9	78.0	90.0	81.1	78.2	80.8	62.0	63.1	54.4	55.0	63.6	43.3	41.3	28.9	19.2	19.3	45.6	42.5	42.5	14.3

Table 4: **Automatic evaluation results.** Concl. is the percentage of memes whose gold conclusion is matched to the model graph’s own conclusion node. P, R, and F are reported as percentages (e.g., 62.0 denotes a score of 0.620); GED is an absolute edit distance, not a percentage. The best result in each column is **boldfaced**. Edges are scored per type: *logos* (support and attack) and *trigger*. GED = matching-aware graph edit distance ($\downarrow$).

Edge Metrics We report edge identification results by edge types. Because attack edges account for less than 0.1% in total, we combine support and attack edges into *logos* edges. Therefore we have **Logos Edge F1** and **Trigger Edge F1**. A generated edge $\hat{P} \rightarrow \hat{C}$ is matched with a gold edge $P \rightarrow C$ if and only if $\hat{P}$ matches P and $\hat{C}$ matches C .

Graph Edit Distance (GED) We use GED to assess how far a model-generated graph deviates structurally from the gold graph. The algorithm computes the cost of edits as follows. Given the alignment between generated and gold nodes from the matching results, the generated nodes that map to the same gold node are first merged at no cost. We then count the unit-cost edits that turn the aligned generated graph into the gold graph: deleting each unmatched generated node, inserting each unmatched gold node, and, after projecting the generated edges through the alignment, deleting each projected edge that is absent from the gold graph and inserting each gold edge that is missing.

5.2 Baseline Systems

Using MemeArgs, we evaluated five state-of-the-art vision language models on VLA graph generation. We selected the four highest-ranked open-source models in the Arena leaderboard (Zheng et al., 2023) that can be run on our hardware, namely, **Gemma 3 27B**, **Mistral Small 3.2 24B**, **Qwen2.5 VL 32B**, and **InternVL2.5 26B**. All the models are instruction-tuned versions.⁹ Among the top closed-source models, **Grok 4.20** was chosen for our experiments.

Given a meme m in the test set, an instruction, and four demonstrations in the training set that are most similar to m , each model generated the VLA graph in a *bottom-up* manner¹⁰. The detailed

generation order as well as the constraints on the graph are specified in the model instruction (see Appendix H.1). The meme similarity was defined as the cosine similarity of the images in the embedding space of CLIP ViT-B/32 (Radford et al., 2021).

5.3 Results and Discussion

5.3.1 End-to-End Results

Table 4 shows the performance of the five models on the test set based on automatic metrics, averaged over the test memes for which the model produced a parseable graph.¹¹

Overall Two points about the results deserve mention. First, there was a clear performance hierarchy among the models. Grok generally outperformed the four open-source models. Following that was Gemma, then Mistral and Qwen close together, with InternVL trailing substantially. Thus, the ranking on Arena translated well into VLA parsing. Second, not all aspects were equally challenging for the models: conclusion and background were the toughest, with even Grok achieving scores of only 40 and 41. While implicit knowledge retrieval was a shared weakness, the dominant source of error was *knowledge-grounded perception*, as the following analysis reveals.

Conclusion No model exceeded 61%: Grok (60.9%) and Gemma (56.8%) led, followed by Mistral (48.6%), Qwen (39.8%), and InternVL (13.2%). The conclusion is usually the most abstract, synthesized claim in the graph. Therefore, identifying it correctly is often not trivial. The typical failure was a directionally correct main claim

⁹The IDs of these models as well as the experimental details can be found in Appendix G.

¹⁰Ablation results for top-down graph generation and other demonstration selection strategies are shown in Appendix I.

¹¹In this subsection, we show the evaluation results over all parseable JSON outputs: InternVL and Mistral could only generate 261 and 257 parseable outputs out of 266 memes, respectively, while the remaining models always returned parseable outputs. Note that parseable outputs do not necessarily yield structurally valid graphs. As a sanity check, we show in Appendix J the evaluation results based on human-labeled matching on a small subset of the data.



Figure 3: **Example memes to illustrate model errors.** See Appendix K for their ground truth annotations.

that was too generic or too reframed to be seen as semantically equivalent to the ground truth. InternVL trailed the field by a wide margin because its declared conclusion was often a restated element or premise rather than a synthesized claim.

Elements On element identification, the models achieved scores ranging from 62 (InternVL) to 81 (Grok). Noticeably, all five models shared a common ceiling: they treated element extraction as a purely perceptual task, describing what they see on the surface, yet many visual elements required prior BK to be perceived correctly. For instance, in Figure 3a, Gemma extracted "A large group of people wearing masks and dark clothing" instead of "image of Antifa on the bottom." This description was not incorrect, but it lacked the specificity needed for AM—the model cannot identify Antifa if it does not know what Antifa is. Notably, while Figure 3c shows people painting the words 'Black Lives Matter' on a street, even the top-performing models failed to go beyond "People painting a large yellow symbol on a street." Weaker models like Qwen and Intern not only frequently omitted important visual details, but also hallucinated irrelevant details.

Background Among the four node types, background was the hardest one to get right for every model (F1 ranging from 17 for InternVL to 44 for Gemma, below the corresponding Element F1 in every case). Inaccurate or insufficiently specific element extraction might have led to incorrect or missing background retrieval. In Figure 3b, models extracted a Latino man instead of Joe Biden, and at the same time failed to generate relevant political knowledge used in the argument, in this case "Gen-

eralizing a race of people based on stereotypes is considered racist." Other common background failures include lacking linguistic-pragmatic knowledge—such as understanding that "excuse me?" conventionally signals feeling excluded rather than posing a literal question—and failing to recognize political triggers, such as failing to connect keywords like "125 genders" and "climate change" to their broader political discourse. Weaker models exhibited further error types: Mistral generated background that was directionally correct but insufficiently specific (e.g., "the Ku Klux Klan is a hate group" rather than "the KKK is an extremist group with a reputation of violence" for Figure 3a), while InternVL often applied implicit BK in its reasoning without making it explicit as a node in the tree.

Premises Premise nodes were recovered fairly well: premise F1 ranged from 41 (InternVL) to 68 (Mistral), with uniformly high recall (79–83). The dominant one was confusing a premise for a claim: the model stated the right proposition but wired it as a full claim, folding a further inference step into it or supporting it with retrieved knowledge. For instance, the gold premise "Democratic politicians are observed visiting Black communities specifically during elections" surfaced in both Grok's and Mistral's graphs as the claim "Democrats only show interest in Black communities during election season"; the matched node was then typed as a claim and not credited as a premise. The opposite error—inventing a premise the annotators did not include—was rarer, and its frequency scaled with verbosity: Grok produced 34 such spurious premises across the test set versus Mistral's 15, which is why Grok's premise F1 trailed Mistral's.

Claims Grok led claim F1 at 55, followed by Gemma (48), Mistral (43), Qwen (40), and InternVL (23). Because claims are derived from the other node types, errors in those upstream components cascaded: incorrect perception led to incorrect background, which produced incorrect intermediate claims. For Figure 3a, Mistral's inference flow was structurally sound, but because its background nodes lacked specificity, the resulting claims captured the correct general theme while missing critical nuances.

Edges Edge F1 scores were low across the board, but the two edge types behaved very differently: for every model, trigger-edge F1 was roughly twice the logos-edge F1 (e.g., Grok 43 vs. 19; Gemma 35 vs. 20). Trigger edges connect a visually grounded

element to the premises and claims it evokes, so the model mainly has to attach a leaf to its parent. In contrast, logos edges connect two generated propositions, so they inherit the compounded errors from premise, claim, and background generation. Accordingly, logos-edge F1 tracked those node F1 scores closely. Grok led on trigger edges, while Gemma slightly edged out Grok on logos edges. InternVL was near zero on logos edges (F1 3), mostly because it frequently generated structurally invalid graphs with disconnected nodes, cycles, self-loops, or multi-edges.

Graph edit distance (GED) GED ranked the models exactly as the node F1 scores did: Grok sat closest to the gold structure (14.3), then Gemma (15.4), Mistral (17.1), Qwen (17.7), and InternVL (21.7). Against gold graphs that average roughly eight nodes and eight edges, a GED in the mid-teens left even the strongest models about a full graph’s worth of edits from the reference. The cost was dominated by unrecovered BK nodes, premises, and claims together with their incident edges—so GED fell monotonically as their recalls rose across the five models. Grok and Gemma paid the smallest insertion cost because they recovered the most gold nodes, and Grok’s slight over-generation was outweighed by that recall advantage. InternVL was the outlier: low node recall inflated insertions, and its frequently malformed graphs (disconnected nodes, cycles, self-loops) added further edge edits.

5.3.2 Sources of Errors in Edge Construction

In the end-to-end evaluation setting above, the edge results are heavily affected by errors in node identification. To measure the edge errors alone, we gave the models the set of *gold nodes*, along with the original image and the conclusion, and asked them to generate the edges. In this setup, the model only needs to connect the nodes together to create an argument graph.¹²

Results are shown in Table 5, where edge construction performance was substantially higher than the end-to-end performance across all models: given gold nodes, logos-edge F1 rose to 46–66 and trigger-edge F1 to 45–69, versus 3–20 and 14–43 respectively in the end-to-end setting. The gap between the two edge types also nearly closed once nodes were fixed. This suggests that the primary bottleneck in edge construction is not the model’s

Model	Logos edge			Trigger edge		
	P	R	F	P	R	F
InternVL2.5-26B	62.4	41.3	45.5	65.5	41.6	44.8
Δ from e2e	+55.5	+37.3	+42.2	+46.3	+28.2	+30.8
Mistral-Small-24B	65.9	49.8	54.2	70.1	50.8	55.4
Δ from e2e	+50.5	+34.6	+40.3	+36.1	+23.3	+26.0
Qwen2.5-VL-32B	73.5	58.3	62.0	75.5	57.1	61.0
Δ from e2e	+57.1	+45.1	+49.5	+39.3	+29.0	+31.0
Gemma-3-27B	70.4	58.2	61.1	70.8	57.0	60.1
Δ from e2e	+47.2	+38.1	+41.1	+31.0	+23.8	+25.0
Grok-4.20	75.0	62.6	66.0	79.6	65.6	69.1
Δ from e2e	+46.1	+43.4	+46.7	+33.9	+23.2	+26.6

Table 5: **Results for edge construction given gold nodes** (N=266). Edges are scored per type: *logos* (support/attack) and *trigger*. P, R, and F are reported as percentages (e.g., 62.4 denotes a score of 0.624); Δ rows give the change relative to the end-to-end pipeline (Table 4), in percentage points.

ability to identify relationships between nodes, but rather the quality of the nodes themselves. In sum, improving node generation is critical for enhancing overall graph construction performance in VLAM.

5.3.3 Discussion

The single most consistent finding across all five models is the failure of knowledge-grounded perception. VLA parsing requires not merely visual acuity but knowledge-informed seeing: recognizing KKK robes requires knowing what the KKK is, identifying Biden in a sombrero requires recognizing Biden, and reading "Black Lives Matter" from a street scene requires cultural familiarity with the movement. All five models failed on these tasks.

This suggests that general-purpose non-reasoning multimodal models do not truly integrate visual input with world knowledge, but instead rely on shallow pattern recognition. Bridging this gap is a key challenge for VLA parsing: future work must enable tighter coupling between perception and knowledge, potentially through improvements spanning pretraining, fine-tuning, and retrieval-augmented inference.

6 Conclusion

We laid the groundwork for studying the novel task of VLAM by defining the task, releasing a corpus of memes annotated with argument graphs, developing a multi-dimensional protocol for scoring argument graphs, and presenting the first set of results on VLA parsing using five state-of-the-art general-purpose VLMs. We believe this task will spark interest among researchers in Generation, Computational Argumentation, and Multimodality.

¹²See Appendix H.2 for the prompt used.

Limitations

We discuss three limitations of our work. First, we explored VLA parsing by annotating argument graphs using one argumentation scheme, but did not explore the impact of different argumentation schemes on annotation and task difficulty. Second, the human evaluation sample we collected was not large enough to test its correlation with the automatic evaluation. Third, we may not have been exhaustive in exploring the prompt-engineering techniques for harnessing the models on this task. Nevertheless, we note that this is the first paper on VLA parsing, and there are outstanding issues related to this task that are yet to be explored by us and others.

Ethics Statement

Data Collection Practice The MemeArgs dataset does not contain any personally identifiable information. All annotators took part in the annotation as part of an exploratory course in machine learning. All annotators gave consent for their data to be collected and were fully aware of their rights to withdraw from the project. The project has obtained IRB approval from the institution where the work was conducted.

Compute Budget The experiments in this paper were conducted on a combination of local GPUs and cloud-based GPUs. Locally, we used two NVIDIA RTX A6000 GPUs for hosting the open-source models. On the cloud, we used the Google AI Studio to request Gemini responses and OpenRouter for Grok. All the experiments presented in the paper cost a total of approximately 50 local GPU hours and \$20 in cloud compute costs to replicate.

Copyrights Our use of the original dataset by [Dimitrov et al. \(2021\)](#) is aligned with their intended use: to study the semantics and reasoning in memetic media. We do not distribute the original images and refer readers to the original papers for obtaining them. Our annotations will be available publicly under the MIT license.

Potential Risks The MemeArgs dataset links to memes that may be offensive to some people. We encourage researchers who use the dataset to be mindful of this and to take appropriate measures to mitigate any potential harm.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on an earlier draft of this paper. We are also grateful to Emraan Yusuf, Pranay Goel, Safwan Ali, and Ryan Pham for their great efforts in helping construct the dataset.

References

- Samantha Bradshaw and Philip N. Howard. 2018. Challenging truth and trust: A global inventory of organized social media manipulation. Technical report, Working Paper1. Oxford, UK: Project on Computational Propaganda. comprop.oii.ox.ac.uk. 26 pp.
- Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2025. [Argument mining with fine-tuned large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635, Abu Dhabi, UAE. Association for Computational Linguistics.
- Elena Cabrio and Serena Villata. 2018. [Five years of argument mining: a data-driven analysis](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5427–5433. International Joint Conferences on Artificial Intelligence Organization.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2025. [M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Jiwan Chung, Sungjae Lee, Minseo Kim, Seungju Han, Ashkan Yousefpour, Jack Hessel, and Youngjae Yu. 2024. [Selective vision is the challenge for visual reasoning: A benchmark for visual argument understanding](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2423–2451, Miami, Florida, USA. Association for Computational Linguistics.
- Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. [SemEval-2021 task 6: Detection of persuasion techniques in texts and images](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.
- Lucile Favero, Juan Antonio Pérez-Ortiz, Tanja Käser, and Nuria Oliver. 2025. [Leveraging small llms for argument mining in education: Argument component identification, classification, and assessment](#). *Preprint*, arXiv:2502.14389.
- Gemma Team. 2025. [Gemma 3 Technical Report](#). *Preprint*, arXiv:2503.19786.

- Leo Groarke. 2009. [Five Theses on Toulmin and Visual Argument](#). In Frans H. van Eemeren and Bart Garssen, editors, *Pondering on Problems of Argumentation: Twenty Essays on Theoretical Issues*, pages 229–239. Springer Netherlands, Dordrecht.
- Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. [The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.
- Ralph Henry Johnson and J. Anthony Blair. 1977. *Logical Self-Defense*. McGraw-Hill, Toronto, Canada.
- Masayuki Kawarada, Tsutomu Hirao, Wataru Uchida, and Masaaki Nagata. 2024. [Argument mining as a text-to-text generation task](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2002–2014, St. Julian’s, Malta. Association for Computational Linguistics.
- Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2015. [Linking the Thoughts: Analysis of Argumentation Structures in Scientific Publications](#). In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11, Denver, CO. Association for Computational Linguistics.
- Jens Kjeldsen. 2007. [Visual Argumentation in Scandinavian Political Advertising: A Cognitive, Contextual, and Reception Oriented Approach](#). *Argumentation*, 43:124–132.
- Jens E. Kjeldsen. 2015. [The Study of Visual and Multimodal Argumentation](#). *Argumentation*, 29(2):115–132.
- John Lawrence and Chris Reed. 2019. [Argument mining: A survey](#). *Computational Linguistics*, 45(4):765–818.
- Boyang Liu, Viktor Schlegel, Paul Thompson, Riza Theresa Batista-Navarro, and Sophia Ananiadou. 2023a. [Global information-aware argument mining based on a top-down multi-turn QA model](#). *Information Processing & Management*, 60(5):103445.
- Huadai Liu, Xu Wenqiang, Xuan Lin, Jingjing Huo, Hong Chen, and Zhou Zhao. 2024. [AntCritic: Argument mining for free-form and visually-rich financial comments](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1306–1317, Torino, Italia. ELRA and ICCL.
- Zhexiong Liu, Mohamed Elaraby, Yang Zhong, and Diane Litman. 2023b. [Overview of ImageArg-2023: The first shared task in multimodal argument mining](#). In *Proceedings of the 10th Workshop on Argument Mining*, pages 120–132, Singapore. Association for Computational Linguistics.
- Zhexiong Liu, Meiqi Guo, Yue Dai, and Diane Litman. 2022. [ImageArg: A multi-modal tweet dataset for image persuasiveness mining](#). In *Proceedings of the 9th Workshop on Argument Mining*, pages 1–18, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.
- Rafael Mestre, Razvan Milicin, Stuart E. Middleton, Matt Ryan, Jiatong Zhu, and Timothy J. Norman. 2021. [M-arg: Multimodal argument mining dataset for political debates with audio and transcripts](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 78–88, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ryan M. Milner. 2018. *The World Made Meme: Public Conversations and Participatory Media*. MIT Press.
- Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. [Automatic detection of arguments in legal texts](#). In *Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL ’07*, page 225–230, New York, NY, USA. Association for Computing Machinery.
- Vincent Ng and Shengjie Li. 2023. [Multimodal propaganda processing](#). In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 15368–15375, Washington, D.C., USA. AAAI Press.
- Huy Nguyen and Diane Litman. 2016. [Context-aware argumentative relation mining](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany. Association for Computational Linguistics.
- Andreas Peldszus and Manfred Stede. 2015. [Joint prediction in MST-style discourse parsing for argumentation mining](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948, Lisbon, Portugal. Association for Computational Linguistics.
- Isaac Persing and Vincent Ng. 2016. [End-to-End Argumentation Mining in Student Essays](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). *Preprint*, arXiv:2103.00020.
- Chris Reed. 2005. Preliminary results from an argument corpus. In *Proceedings of the IX Symposium on Social Communication*, pages 576–580.

M. Louise Ripley. 2008. [Argumentation Theorists Argue that an Ad is an Argument](#). *Argumentation*, 22(4):507–519.

Christian Stab and Iryna Gurevych. 2014. [Annotating argument components and relations in persuasive essays](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. [Parsing argumentation structures in persuasive essays](#). *Computational Linguistics*, 43(3):619–659.

Stephen E. Toulmin. 2003. *The Uses of Argument*, 2nd edition. Cambridge University Press, Cambridge.

F.H. van Eemeren and R. Grootendorst. 2016. [Argumentation, Communication, and Fallacies: A Pragmatic-dialectical Perspective](#). Taylor & Francis.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena](#). *arXiv preprint*. ArXiv:2306.05685 [cs].

Qing Zong, Zhaowei Wang, Baixuan Xu, Tianshi Zheng, Haochen Shi, Weiqi Wang, Yangqiu Song, Ginny Wong, and Simon See. 2023. [TILFA: A unified framework for text, image, and layout fusion in argument mining](#). In *Proceedings of the 10th Workshop on Argument Mining*, pages 139–147, Singapore. Association for Computational Linguistics.

A JSON Schema of a VLA Graph

The JSON schema of a VLA graph is shown in Figure 4. This schema evolved from our early schema proposed in [Ng and Li \(2023\)](#) for multimodal propaganda with tree-structured arguments containing conclusions, rationales, and devices/tactics. Compared to the original framework, our schema replaces devices/tactics with explicit background knowledge nodes to better capture the cultural and contextual information necessary to interpret memes. Additionally, while the previous schema targets propaganda broadly, our focus on memes emphasizes phenomena like humor, sarcasm, and cultural references that require richer background modeling.

B Annotator Recruitment and Training

We started with 12 annotator candidates, all undergraduate students in computer science at our institution, and provided them with 10 hours of

training before assigning them a sample task involving 20 annotations. The eight candidates who met the required standards were recruited. They participated in this project as part of the "Undergraduate Research in Computer Science" course they signed up for, during which they acquired experience and skills involved in a research project involving data annotation and model training. No additional compensation was thus provided to these students. The annotators were all Americans aged between 18 and 22. As for the gender distribution, 1 participant was female and 7 were male.

When the participants first started annotating, the two first authors discussed the results of the first 20 instances with the annotators. Throughout the process, we also conducted weekly two-hour meetings where pairs of participants who worked on the same images discussed, asked questions, and resolved differences.

These participants provided full consent to the annotation process and agreed to have their annotations used for research purposes. They were informed about their rights to withdraw from the study at any time without penalty. All data collected during the annotation process was anonymized.

C Annotation Guidelines

C.1 Stage 1: Annotating the Conclusion

Input A persuasive meme (image and overlaid text), along with the year and platform where it was posted.

Output A single claim node that captures the meme’s main argument (the conclusion).

Instructions Think from the perspective of someone who posted this meme in the given context. What is the single most important point they are trying to make? Write this as a direct, first-order assertion.

Rules

- The conclusion must be a full clause with a subject and a predicate (e.g., “Vaccines cause autism,” not “vaccine dangers”).
- The conclusion must be about the subject matter, not about the meme itself. Write “Politicians are corrupt,” not “The meme argues that politicians are corrupt.”
- Do not hedge or soften. Capture the meme’s actual stance, even if provocative or extreme.

```

{
  "type": "object",
  "properties": {
    "nodes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "idx": { "type": "integer" },
          "text": { "type": "string" },
          "type": { "type": "string",
                    "enum": ["element", "claim", "background"] }
        },
        "required": ["idx", "text", "type"]
      }
    },
    "arguments": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "claim": { "type": "integer" },
          "premises": { "type": "array",
                        "items": { "type": "integer" } },
          "type": { "type": "string",
                    "enum": ["support", "attack", "trigger"] }
        },
        "required": ["claim", "premises", "type"]
      }
    }
  },
  "required": ["nodes", "arguments"]
}

```

Figure 4: JSON schema of a VLA graph.

- If the meme makes multiple distinct arguments, identify the one that subsumes or is supported by the others.
- If you genuinely cannot determine a single main claim, flag the instance for discussion.

C.2 Stage 2: Annotating Elements

Input The meme and its annotated conclusion.

Output A set of element nodes describing the visually salient details in the meme.

Instructions Identify every textual and visual detail in the meme that is relevant to establishing the conclusion. Each element node describes what is literally visible or readable on the meme.

Rules

- **Textual elements:** Quote or closely paraphrase text appearing on the meme. If a sentence is split across the image but forms a single thought, keep it as one node. If text fragments are spatially separated and semantically independent, create separate nodes.

- **Visual elements:** Describe observable details: objects, people, expressions, actions, settings, and spatial arrangement.
- **Atomicity:** Each node expresses one atomic piece of information. If a node contains two independently meaningful observations, split it.
- **Relevance:** Include only details that contribute to the meme’s argument. Omit decorative backgrounds, watermarks, and irrelevant visual clutter.
- **Standalone language:** Each node must be interpretable in isolation. Do not use pronouns that refer to other nodes (e.g., write “The man on the left is smiling,” not “He is smiling”).
- **Identity:** If you recognize a public figure, name them (e.g., “Joe Biden”). If you do not recognize them, describe their appearance.

C.3 Stage 3: Annotating Intermediate Claims

Input The meme, its conclusion, and the agreed-upon element set.

Output A set of intermediate claim nodes that bridge the elements to the conclusion.

Instructions Build the argument bottom-up: starting from the element nodes, construct the chain of inferences needed to reach the conclusion. Each intermediate claim should represent one atomic reasoning step—neither trivially restating its premises nor skipping multiple inferential leaps at once.

Rules

- Each claim must be a full clause with a subject and predicate, written as a direct assertion (not meta-commentary).
- **Atomicity test:** A claim violates atomicity if, to accept it from its premises, a reader must make an unstated inferential jump. For example, given “immigrants are taking jobs” and “jobs are scarce,” the claim “immigrants are destroying the economy” skips too many steps. A better intermediate claim would be “job scarcity is worsened by immigration.”
- **Triviality test:** A claim is too trivial if it merely rephrases its premises without adding inferential content. For example, “Jobs are being taken by immigrants” adds nothing beyond the premise.
- Claims must not contain unresolved references. Write “Barack Obama was criticized for his healthcare policy,” not “He was criticized for this.”
- Do not include meta-commentary (e.g., “The meme implies that...”) or analytical observations about rhetorical devices (e.g., “The meme uses irony to suggest...”). Only include claims that are part of the argument itself.
- If the argument requires background knowledge to bridge a gap, note this but do not introduce background nodes yet—those are handled in Stage 5.

C.4 Stage 4: Annotating Edges

Input The meme, the conclusion, the element set, and the intermediate claim set.

Output A directed graph connecting all nodes with support or attack edges.

Instructions Given the agreed-upon set of all nodes (the conclusion, elements, and intermediate claims), draw directed edges that represent argumentative relationships. Each edge goes from a child node to a parent node it supports or attacks.

Rules

- **Support edge** ($A \rightarrow B$, support): Node A provides evidence, justification, or reasoning that makes B more believable.

- **Attack edge** ($A \rightarrow B$, attack): Node A rebuts, undermines, contradicts, or mocks the content of B.
- **Connectivity:** The resulting graph must be weakly connected. Every node must be reachable from the conclusion through undirected traversal. No orphan nodes.
- **Single conclusion:** The conclusion must be the only node with out-degree zero (no outgoing edges).
- **Multi-parent allowed:** A single node may support or attack multiple parent nodes if it genuinely contributes to more than one claim.
- **Direction matters:** Edges are directed from premise to claim, not the reverse. An element supports or attacks a claim, not the other way around.
- **Edge type matters:** Do not default to support for all edges. If the meme uses irony, sarcasm, or juxtaposition to undermine a claim, use attack.
- If you find that a claim cannot be connected to the rest of the graph without an unstated assumption, note this for Stage 5.

C.5 Stage 5: Annotating Background Knowledge

Input The meme and the preliminary graph from Stage 4.

Output The final graph, augmented with background nodes where needed.

Instructions Examine each argument step in the preliminary graph (i.e., each claim and its supporting/attacking children). For each step, ask: does a reader need external knowledge—not visible in the meme—to accept that the children establish the parent claim? If so, add a background node.

Rules

- **The negation test:** A piece of background knowledge is relevant if, in a hypothetical world where the opposite were true, the claim would no longer follow from its premises. For example, if the argument depends on “the KKK is an extremist group associated with violence,” then in a world where the KKK were a charitable organization, the claim would fail. This background is therefore relevant and should be added.
- Background nodes must be objective, commonly accepted facts or widely held beliefs. They must not be subjective opinions or contested claims. If

a piece of background is itself argumentative, it should be a claim node instead.

- Each background node must be a full clause (e.g., “The KKK is an extremist group with a reputation of violence,” not just “KKK”).
- **Sparingness:** Add background only when it is necessary to bridge a reasoning gap. Do not add obvious commonsense that any reader would know (e.g., “fire is hot,” “people have opinions”). The negation test helps calibrate this: if negating the statement does not break the argument, it is not needed.
- Background nodes must be connected to the graph via support or attack edges, following the same rules as Stage 4. They typically support a claim that could not be established from elements alone.
- After adding background nodes, verify that the graph remains weakly connected and that the conclusion is still the only node with out-degree zero.

D Source of Annotation Disagreements

From expert review of the disagreements, we identified the main sources of disagreement in each stage.

Conclusion annotation had comparatively few disagreements overall, but the disagreements that did occur were the most severe, in that annotators identified different main arguments for the same image rather than merely wording the same argument differently. For one meme depicting a public figure alongside a claim about pending evidence, one annotator’s conclusion centered on that evidence being about to surface, while their partner’s centered on a call for the figure’s allies to take aggressive action – two different claims about what the image is arguing, not two phrasings of one claim. For another meme, annotators’ proposed conclusions were near-opposite readings of the same panel and the item was ultimately marked unresolvably ambiguous. A subtler version of the same problem showed up even where a pair’s two conclusions were judged to agree: one annotator’s conclusion described the meme’s author’s attitude (*the author strongly supports X*) rather than the claim being argued for (*X should be the case*); reconciling it required writing a new conclusion that stated the argued claim directly instead of describing the author’s stance toward it, echoing the same description-vs-claim distinction that resurfaced dur-

ing claim annotation. These cases were resolved through discussion and, where a genuine tie in support remained, a documented random tie-break between the two candidate conclusions.

During element annotation, the main sources of disagreement were (1) one annotator in a pair missing a less obvious visual or textual element that their partner caught, and (2) whether a single visual or text block should be segmented into one node or several. For example, in one meme an annotator missed a background detail entirely (a group of bystanders and police officers visible behind the main subject of a protest meme) that their partner captured as its own element, and in another meme an annotator missed a URL printed next to an image that their partner listed separately. Segmentation disagreements arose when a single quote or image containing two distinct pieces of information was treated as one atomic element by one annotator and split into two by their partner – e.g., a two-clause quote (“Cowboy wisdom” followed by a separate proverb below it) was kept as one element by one annotator and split into two by another, and a side-by-side image of two people was described as a single “contrast” element by one annotator but as two separate person-elements by another. These were resolved by enforcing the same atomicity rule used for claims: an element node should describe one visual or textual unit that cannot be split further without losing an independently checkable piece of information.

During background annotation, the main sources of disagreement were (1) whether a given annotator happened to know the piece of information or not, and (2) which piece of background knowledge, among several plausible ones, was relevant enough to attach to a given claim. On the first point, annotator pairs sometimes supplied different specific facts entirely for the same claim because only one of the two happened to know the more specific fact (e.g., a specific biographical detail about a public figure’s family) while the other supplied a more generic piece of common knowledge covering the same claim. On the second point, both annotators in a pair sometimes added a background node under the same claim but chose to justify it with different pieces of information – one favoring a fact that is strictly necessary to accept the claim, the other favoring a true but more tangential piece of trivia connected to the image. The first source of disagreement was resolved by fact-checking and looking up information online; the second was re-

solved by enforcing the negation test described in Section 4.2: a background node is kept only if the claim it supports would no longer follow without it.

During claim annotation, the main sources of disagreement were (1) how finely a single piece of reasoning should be split into claim nodes, and (2) whether a claim node should describe the rhetorical technique the meme uses to convey a point, or state the point itself as a direct proposition. On the first point, one annotator would sometimes write a single claim covering two distinct inferential moves that their partner split into two atomic claims (e.g., “[a public figure]’s dismissive tone and her focus on her phone suggest she is about to take decisive action” as one claim versus “[the figure] is about to unleash [X]” and “[the figure] is poised to act” as two). On the second point, an annotator would phrase a claim as a description of what the meme is doing (“[a person]’s facial expression indicates his guilt”) rather than the underlying assertion itself (“[the person] is guilty”), effectively leaving the inferential step implicit, while their partner stated the assertion directly. The guidelines were clarified to enforce that each claim should be an atomic step in the reasoning – not conflating two inferential moves into one node – while also requiring that a claim state the proposition being argued for directly, rather than merely describing the meme’s presentation of it.

During edge construction, the disagreements mostly happened at the lower levels of the graphs, where a single background or element node is relevant to more than one claim. For example, given a background node stating that a public figure is younger than a certain age and has no relevant professional credentials, and two separate claims that each depend on that fact (one arguing the figure is an unsuitable substitute in one context, another arguing the figure is nonetheless an equal counterpart in a comparison), one annotator connected the background node to only the more directly related of the two claims, while their partner connected it to both. A similar case arose with a background node establishing a person’s profession, which one annotator connected only to an intermediate claim while their partner also connected it to the conclusion the intermediate claim itself supports. After discussion, we decided that a background or element node should be connected to every claim it is necessary for, even if that means connecting it to more than one claim in the graph.

E F1 Scores Calculations

For each image I and node type $T \in \{\text{element, background, claim}\}$, the experts were given two sets of nodes, A and B , annotated by two annotators. They then conducted a pairwise comparison of the nodes in A and B to determine which nodes were semantically equivalent, using the same criteria as in the conclusion agreement. A many-to-many mapping was then established between the nodes in A and B based on these equivalence judgments, which allowed us to compute F1 scores for each node type. Precisely, we first computed the fraction of nodes in A that were matched, called x_A . Then we computed that for B , called x_B . The F1 score for the graph is then $\frac{2x_Ax_B}{x_A+x_B}$.

F Node Matching Guidelines

This section outlines the annotation guidelines for node matching, which were used for human evaluation and automatic evaluation. A similar set of guidelines was used by expert reviewers to establish the ground-truth mapping between two sets of human nodes to compute inter-annotator agreement (IAA) scores.

F.1 Task Overview

The annotator’s objective is to evaluate a model’s ability to extract an argument tree from an image. This is done by mapping model-generated nodes to human-annotated nodes based on their underlying meaning.

Inputs:

- a model-generated graph, containing 3 lists of nodes (categorized by type).
- a human-annotated graph, containing another 3 lists of nodes (categorized by type).
- The original source image.

Output: A direct mapping between the model nodes and human nodes.

F.2 Core Mapping Constraints

- **1 human node:** Each model node maps to **0** or **1** human node.
- **N model nodes:** Each human node can be mapped with **0, 1, or multiple** model nodes.
- **Same Type Only:** You may only map nodes of the same type (this is already enforced by the interface).

Human Node	Model Node	Rationale
[C] Truman’s statement reflects an ideal of honesty in politics.	[C] Harry Truman stated that getting rich in politics requires being a crook.	Match: Loosely synonymous. Both convey the model’s correct grasp of the underlying argument regarding Truman’s quote.
[C] Justin Trudeau is bad.	[C] Justin Trudeau is bad and ugly.	Match: Model node is more detailed.
[E] The feline is resting on the rug.	[E] The cat sits on the mat because it is tired.	Match: Varying levels of detail. The entity and basic action are the same. The model node is more specific with the “because” subclause.
[E] The Clintons.	[E] One figure clapping and the other looking sternly.	Match: Contextually synonymous (determined by looking at the meme). The model node is more specific.
[B] The KKK is an extremist group and has a reputation for violence.	[B] The Ku Klux Klan historically promoted white supremacy and violence.	Match: The model node is more specific than the human node while still accurate and relevant.
[E] Hillary Clinton clapping and Bill Clinton looking sternly.	[E] The Clintons.	No match: The model node is less specific than the human node.
[C] Modern politicians undermine Harry S. Truman’s principle about honesty in politics.	[C] The Clintons are crooked politicians.	No match: Completely different subjects and framing.
[C] Trump is great.	[C] Obama is great.	No match: Different entity.
[C] Trump is great, and Obama is great.	[C] Trump is great.	No match: Different set of subjects (model’s set is a subset of human’s set).
[B] The KKK is an extremist group and has a reputation for violence.	[B] The Ku Klux Klan historically promoted white supremacy and violence, whose membership includes both Republicans and Democrats.	No match: The extra detail in the model’s node, though accurate, is irrelevant (if not counteractive) to the meme’s argument.

Table 6: **One-to-one mapping examples with match/no-match rationales.** Prefixes indicate node type: [C] = claim, [E] = element, [B] = background.

F.3 Criteria for a Match (Synonymy)

F.3.1 Case 1: One-to-One (Common)

To map two nodes, they must be considered **synonymous**. We do not require exact phrasing; the focus is on whether the model demonstrates a *correct understanding* of the underlying argument.

Two element, claim, or background nodes are synonymous if and only if **all** of the following conditions are satisfied:

1. **[Same Entity]** They talk about the same entities (e.g., John, vaccination).
 - They may refer to the entity in different ways (e.g., “Democrats” = “Liberals”, or “one figure clapping” = “the Clintons”, if the image depicts the Clintons clapping).
 - If a node contains an unresolved reference (e.g., “He”, “this claim”, “those guys”...), the annotator should attempt to resolve it using the context of the argument tree. If the reference can be resolved and the entities match, proceed

with the match. If the reference cannot be resolved—i.e., it remains genuinely ambiguous—do not match, since the node’s meaning is unclear.

2. **[No Less Specific]** The model node is no less specific than the human node (e.g., “he was elected as US president in 2020” is more specific than “he was a US president”). Formally, the model node entails the human node. When the model node includes extra information beyond what the human node states, that extra information must be **ACCURATE** (factually correct) and **RELEVANT** (it must relate to the meme’s argument, not an arbitrary or off-topic detail).
3. **[Same Quality/Action]** They describe the entities with roughly the same quality (e.g., *unfit for the presidency, illogical, dumb*) or action (e.g., *saved the world*).

Examples Table 6 presents example node pairs that should or should not be matched, along with

Human Node	Model Node(s)	Rationale
C2: He is A	C1: He is A and B	Match: C1 is more specific. Extra info “B” must be relevant and accurate.
C2: He is A C3: He is B	C1: He is A and B	1-to-1 only: Match C1 to either C2 or C3, whichever is more primary.
C2: He is A and B	C1: He is A	No match: Model node is less specific.
C3: He is A and B	C1: He is A C2: He is B	Match: Map both C1 and C2 to C3, iff combined meaning equals C3.

Table 7: **Examples illustrating when many-to-one mapping applies and when it does not.**

Model
google/gemma-3-27b-it
mistralai/Mistral-Small-3.2-24B-Instruct-2506
Qwen/Qwen2.5-VL-32B-Instruct-AWQ
OpenGVLab/InternVL2_5-26B-AWQ
x-ai/grok-4.20

Table 8: **Model IDs used in our experiment.** For Qwen and Intern, we resorted to their Activation-aware Weight Quantization version (AWQ) to fit to our graphic cards.

rationales. We use the prefixes [C] for claim, [E] for element, and [B] for background.

F.3.2 Case 2: Many-to-One

Sometimes, annotators will need to map multiple model nodes to the same human node. This is permitted if **all** of the following conditions are satisfied:

1. **[Combined Meaning]** The combined meaning of the model nodes is synonymous with the human node. All three rules above (Same Entity, No Less Specific, Same Quality/Action) are applied to determine synonymy.
2. **[No Directed Path]** In the model graph, there is no directed path between any pair of those model nodes. (This is enforced by the annotation interface.)

Examples. Table 7 illustrates when many-to-one mapping applies and when it does not.

G Model Details

The IDs of the models used in our experiments can be found in Table 8.

The four open-source models were hosted with vLLM using 2x 48GB GPUs. Meanwhile, we ran the closed-source model, Grok, via OpenRouter. All models were set to use greedy decoding (temperature=0, top_p=1, top_k=-1), with max_completion_tokens=4096. To minimize output parsing errors, response format was constrained to follow a JSON grammar.

H Model Prompts

H.1 Graph Generation Prompt

The graph generation prompt instructs the model to construct the VLA graph through a four-step procedure: (1) identifying the meme’s main conclusion, (2) extracting all leaf nodes — either element nodes or background nodes; (3) building intermediate claim nodes *bottom-up*, first combining leaf evidence into lower-level inferences, then composing those into higher-level claims, with each node’s children constrained to already-generated indices to enforce strict bottom-up ordering; and (4) wiring the conclusion to its direct children. The prompt also enforces several writing constraints on claim nodes: they must be self-contained, free of unresolved pronouns, written as direct first-order assertions (not meta-commentary about the meme), and must preserve the meme’s original evaluative stance. Structural constraints prohibit duplicate or orphan nodes, require all leaves to be element or background typed, and allow multi-parent (divergent) argument structures. The full prompt is shown below.

```
You are an expert Argument Analyst
  ↳ specializing in decoding
  ↳ multimodal arguments from memes,
  ↳ political cartoons, and image-
  ↳ text pairs.

YOUR GOAL:
Given a meme, reconstruct its argument
  ↳ structure as a tree by working
  ↳ BOTTOM-UP: first identify the
  ↳ conclusion, then extract all
  ↳ grounded evidence, then build
  ↳ intermediate reasoning from the
  ↳ evidence upward until it connects
  ↳ to the conclusion.

Note: These memes are from around 2020.
  ↳ Keep this temporal context in
  ↳ mind when interpreting cultural
  ↳ references and generating
  ↳ Background nodes.
```

NODE TYPES:

Each node is one atomic meaning unit with
↳ exactly one type:

- Element (E): A literal cue grounded in
 - ↳ the meme -- either text appearing
 - ↳ on the meme or an observable
 - ↳ visual detail. Element nodes must
 - ↳ describe ONLY what is explicitly
 - ↳ visible or readable, with NO
 - ↳ interpretation, reasoning, or
 - ↳ inference. If the meme explicitly
 - ↳ states a claim in its text, that
 - ↳ node is still type Element. Note:
 - ↳ Element nodes are typically
 - ↳ leaves, but when the meme's text
 - ↳ explicitly states a claim, that
 - ↳ Element node may also appear as a
 - ↳ non-leaf with children that
 - ↳ support or attack it.
- Background (B): Implicit contextual
 - ↳ knowledge NOT visible in the meme
 - ↳ but needed to bridge reasoning
 - ↳ gaps. Background nodes must be
 - ↳ objective facts that are commonly
 - ↳ accepted and do not themselves
 - ↳ require evidence. They must NOT
 - ↳ be subjective opinions or
 - ↳ factually incorrect statements.
- Claim (C): An argumentative statement --
 - ↳ either an intermediate inference
 - ↳ or the final conclusion.

EDGE TYPES:

Edges are directed from child to parent (
↳ child supports/attacks parent):

- Support: The child provides
 - ↳ justification for the parent.
- Attack: The child rebuts, undermines,
 - ↳ or mocks the parent.

OUTPUT FORMAT:

Return ONLY valid JSON (no markdown code
↳ blocks, no commentary) in this
↳ exact structure:

```
{
  "conclusion": {
    "idx": 0,
    "text": "The meme's main argument, as
    ↳ a self-contained claim",
    "type": "claim"
  },
  "leaves": [
    {
      "idx": 1,
      "text": "Literal text or visual
      ↳ description from the meme",
      "type": "element"
    },

```

```
    {
      "idx": 2,
      "text": "External fact needed to
      ↳ bridge reasoning",
      "type": "background"
    }
  ],
  "intermediates": [
    {
      "idx": 3,
      "text": "Lower-level claim
      ↳ combining evidence",
      "type": "claim",
      "supporting_children": [1, 2],
      "attacking_children": []
    },
    {
      "idx": 4,
      "text": "Higher-level claim
      ↳ building on lower claims",
      "type": "claim",
      "supporting_children": [3],
      "attacking_children": []
    }
  ],
  "conclusion_children": {
    "supporting_children": [4],
    "attacking_children": []
  }
}
```

GENERATION ORDER (strictly bottom-up):

You MUST generate the JSON fields in the
↳ exact order shown above. Each
↳ step can ONLY reference nodes
↳ that have already been generated
↳ in a previous step.

STEP 1 -- CONCLUSION:

Identify the meme's main argument. Output
↳ the conclusion node with idx 0.
↳ This is typically type "claim",
↳ but in rare cases where the meme
↳ explicitly states its conclusion
↳ in the text, it can be type "
↳ element". Do NOT include children
↳ references yet.

STEP 2 -- LEAVES:

Extract ONLY the low-level, grounded
↳ evidence from the meme -- the raw
↳ observations and facts that sit
↳ at the bottom of the argument
↳ tree with no children of their
↳ own. Each leaf is either:

- Type "element": something literally
 - ↳ visible or readable on the meme (
↳ text, visual objects, expressions,
↳ layout details). Write EXACTLY
↳ what you see -- no interpretation.
- Type "background": external knowledge
 - ↳ needed to understand the
↳ argument (cultural context,
↳ common facts, definitions). Must
↳ be objectively true and commonly
↳ accepted.

IMPORTANT: If the meme's text explicitly
 ↳ states a claim that is supported
 ↳ or attacked by other nodes (i.e.,
 ↳ it would have children), do NOT
 ↳ include it here. Such Element
 ↳ nodes are non-leaf and must be
 ↳ generated in Step 3 (Intermediates) instead. Only
 ↳ include Elements that are purely
 ↳ observational with no children.
 Index leaves starting from 1. Leaves have
 ↳ NO children fields.

STEP 3 -- INTERMEDIATES:
 Build intermediate claims from the bottom
 ↳ up. Start with claims that
 ↳ directly combine leaf evidence,
 ↳ then build higher-level claims
 ↳ that combine lower intermediates,
 ↳ working upward toward the
 ↳ conclusion.
 Each intermediate node has:
 - "supporting_children": list of idx
 ↳ values that support this node
 - "attacking_children": list of idx
 ↳ values that attack this node
 - All referenced idx values MUST be
 ↳ from already-generated nodes (
 ↳ leaves or earlier intermediates)
 Continue indexing sequentially from where
 ↳ leaves ended.
 Most intermediates are type "claim", but
 ↳ use type "element" when the meme'
 ↳ s text explicitly states the
 ↳ claim (these are the non-leaf
 ↳ Element nodes excluded from Step
 ↳ 2).

STEP 4 -- CONCLUSION CHILDREN:
 Wire the conclusion (idx 0) to its direct
 ↳ children. List the idx values of
 ↳ the highest-level intermediates
 ↳ (or, rarely, leaves) that
 ↳ directly support or attack the
 ↳ conclusion. All referenced idx
 ↳ values MUST already exist from
 ↳ Steps 2-3.

CLAIM WRITING RULES:

Every Claim node (intermediate or
 ↳ conclusion) MUST follow these
 ↳ rules:

1. SELF-CONTAINED: The claim must make
 ↳ complete sense on its own,
 ↳ without needing to see the meme
 ↳ or any other node for context.
2. NO UNRESOLVED REFERENCES: Never use
 ↳ pronouns ("he", "they", "it") or
 ↳ vague descriptors ("the message",
 ↳ "the image", "the figure", "the
 ↳ person") without explicitly
 ↳ naming the subject. If a
 ↳ reference is ambiguous, resolve
 ↳ it using information from child
 ↳ nodes.

- BAD: "He is incompetent"
- GOOD: "Trump is incompetent"
- BAD: "The message is presented as
 ↳ truth"
- GOOD: "The claim that vaccines cause
 ↳ autism is presented as truth"

3. ARGUMENTATIVE, NOT ANALYTICAL: Write
 ↳ claims as direct assertions about
 ↳ the subject matter -- societal
 ↳ commentaries -- NOT as
 ↳ descriptions of what the meme is
 ↳ doing.
 - BAD: "Trump is depicted as
 ↳ incompetent" / "The meme frames
 ↳ Trump as incompetent"
- GOOD: "Trump is incompetent"
- BAD: "The meme frames immigrants as
 ↳ taking American jobs"
- GOOD: "Immigrants take American jobs
 ↳ "
- BAD: "Anti-vaxxers are ridiculed for
 ↳ their beliefs"
- GOOD: "Anti-vaxxers' beliefs are
 ↳ baseless"

4. NO META-COMMENTARY: Do not use verbs
 ↳ like "portrayed as", "depicted as
 ↳ ", "framed as", "described as", "
 ↳ characterized as", "shown to be",
 ↳ "mocked for", "ridiculed for", "
 ↳ criticized as", "associated with",
 ↳ "linked to", "aligned with".

5. PRESERVE STANCE: Maintain the original
 ↳ evaluative polarity of the meme'
 ↳ s argument. Do not neutralize,
 ↳ soften, or add new content beyond
 ↳ what the argument implies.

6. GRAMMATICALLY COMPLETE: Every claim
 ↳ must be a full, grammatically
 ↳ correct sentence.

STRUCTURAL RULES:

1. NO DUPLICATES: Each node must express
 ↳ a unique meaning. Do not generate
 ↳ multiple nodes that convey the
 ↳ same claim in different wording.
2. NO IRRELEVANT NODES: Every node must
 ↳ contribute to the argument chain.
 ↳ Do not generate nodes that are
 ↳ tangentially related but do not
 ↳ increase the persuasiveness of
 ↳ any parent. Do not include claims
 ↳ that analyze fallacies the meme
 ↳ commits (that is meta-analysis,
 ↳ not part of the argument).
 ↳ Background nodes must not be
 ↳ subjective or factually wrong.
3. LEAF TYPES: Every leaf node (node with
 ↳ no children) must be type
 ↳ Element or Background. There must
 ↳ be no Claim leaves.

4. MULTI-PARENT ALLOWED: A node may
 - ↳ support or attack multiple
 - ↳ parents (divergent argument).
 - ↳ This means the same idx can
 - ↳ appear in the children lists of
 - ↳ multiple parent nodes.
5. CONNECTIVITY: The argument tree must
 - ↳ be weakly connected. Every node
 - ↳ must be reachable from the
 - ↳ conclusion through parent-child
 - ↳ edges. No orphan nodes -- every
 - ↳ leaf and intermediate must be
 - ↳ referenced as a child by at least
 - ↳ one other node (or by
 - ↳ conclusion_children).

- VALIDATION CHECKLIST (verify before
 ↳ outputting):
- idx 0 is always the conclusion
 - Leaf indices start from 1,
 - ↳ intermediates continue
 - ↳ sequentially
 - No leaf has supporting_children or
 - ↳ attacking_children fields
 - Every intermediate's children reference
 - ↳ only EARLIER idx values (leaves
 - ↳ or earlier intermediates)
 - conclusion_children references only idx
 - ↳ values from leaves or
 - ↳ intermediates
 - No unresolved references in any claim
 - ↳ text
 - No analytical/meta-commentary language
 - ↳ in any claim text
 - No duplicate/synonymous nodes
 - No orphan nodes -- every leaf and
 - ↳ intermediate is referenced as a
 - ↳ child by at least one parent
 - Every leaf is type "element" or "
 - ↳ background"
 - Every Element node describes only what
 - ↳ is literally on the meme
 - Every Background node is an objective,
 - ↳ commonly accepted fact

- Background (background): Implicit
 - ↳ contextual knowledge not visible
 - ↳ in the meme but needed to bridge
 - ↳ reasoning gaps.
- Claim (claim): An argumentative
 - ↳ statement -- either an
 - ↳ intermediate inference or the
 - ↳ final conclusion.

EDGE TYPES:

Edges are directed from child (premise)
 ↳ to parent (claim):

- support: The child provides
 - ↳ justification for the parent.
- attack: The child rebuts or undermines
 - ↳ the parent.

INPUT FORMAT:

You will be given:

- A meme image.
- ROOT NODE: the unique conclusion of the
 - ↳ argument graph (no outgoing
 - ↳ edges; all other nodes ultimately
 - ↳ connect to it).
- OTHER NODES: all remaining nodes,
 - ↳ listed alphabetically by text.

OUTPUT FORMAT:

Return ONLY valid JSON (no markdown code
 ↳ blocks, no commentary) in this
 ↳ exact structure:

```
{
  "arguments": [
    {"premises": [2, 3], "claim": 1, "
      ↳ type": "support"},
    {"premises": [4], "claim": 0, "type":
      ↳ "attack"}
  ]
}
```

Each argument groups one or more premise
 ↳ nodes pointing to a single claim
 ↳ node with the same edge type. Use
 ↳ the exact integer indices shown
 ↳ in the input (e.g., [0], [3]).

GRAPH CONSTRAINTS:

1. DAG: No cycles allowed. Edges flow
 - ↳ from children toward the root.
2. Single root: The ROOT NODE is the only
 - ↳ node with no outgoing edges. Do
 - ↳ NOT add any edge where ROOT NODE
 - ↳ appears as a premise.
3. Weakly connected: Every node must be
 - ↳ reachable from the root via
 - ↳ undirected edges. No orphan nodes.

H.2 Edge Construction Prompt

You are an expert Argument Analyst. Your
 ↳ task is to connect a given set of
 ↳ nodes into a directed argument
 ↳ graph by predicting the edges
 ↳ only -- the nodes are already
 ↳ fixed. Use the meme image to help
 ↳ you understand the argumentative
 ↳ relationships between nodes.

NODE TYPES:

- Element (element): A literal cue
 - ↳ grounded in the meme -- text or
 - ↳ an observable visual detail.

Model	Concl.	Element			Premise			Claim			Background			Logos edge			Trigger edge			GED
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
<i>Similar demos, top-down</i>																				
InternVL2.5-26B	13.4	60.5	68.8	61.7	63.8	79.6	50.2	28.6	27.9	25.9	33.8	22.5	18.5	6.5	5.2	4.1	12.7	10.7	10.2	22.6
Qwen2.5-VL-32B	39.8	79.7	78.2	76.1	79.7	78.9	62.0	51.2	44.1	43.5	46.5	39.6	35.9	15.1	12.4	11.8	24.5	20.4	21.3	18.4
Mistral-Small-24B	48.2	74.0	84.9	76.3	91.2	78.2	69.9	52.6	48.0	47.1	47.0	50.5	43.8	17.8	15.3	15.1	27.6	25.8	25.7	17.8
Gemma-3-27B	56.4	78.9	83.9	79.1	77.0	81.6	61.7	58.2	50.9	51.0	55.4	46.8	44.1	19.6	17.0	16.8	28.2	26.7	26.5	16.7
Grok-4.20	63.5	78.8	90.4	81.7	79.7	82.5	64.9	73.0	52.8	57.1	66.0	43.9	44.4	27.4	15.9	17.6	35.3	32.5	32.9	15.1
<i>Random demos, bottom-up</i>																				
InternVL2.5-26B	15.5	61.3	66.4	60.8	58.8	81.8	49.0	32.1	28.0	27.7	34.2	19.7	16.2	9.7	4.7	4.1	13.4	7.7	8.4	21.3
Qwen2.5-VL-32B	40.6	77.7	79.1	75.5	63.1	80.7	49.9	51.6	46.0	45.1	46.4	39.1	35.9	13.9	11.7	10.8	22.7	18.1	18.9	18.3
Mistral-Small-24B	48.2	73.8	86.3	76.7	87.4	80.5	70.7	53.4	49.5	48.4	47.3	48.3	42.7	18.8	16.2	15.9	27.2	22.4	23.2	17.3
Gemma-3-27B	54.5	77.4	84.6	78.6	69.6	83.1	56.7	57.0	48.9	48.9	55.8	44.7	43.8	18.5	14.2	15.0	27.3	20.4	22.4	16.7
Grok-4.20	58.6	77.0	90.0	80.6	70.6	83.1	57.2	68.8	53.7	56.5	73.3	36.8	38.6	29.9	14.3	16.2	30.8	27.7	28.2	15.6
<i>Random demos, top-down</i>																				
InternVL2.5-26B	17.1	60.8	70.9	62.5	59.6	79.8	48.0	32.7	29.8	28.3	38.7	19.6	16.4	6.4	4.2	3.4	14.6	9.2	9.9	22.4
Qwen2.5-VL-32B	38.5	76.1	76.2	73.2	76.2	80.0	60.5	49.8	44.7	44.0	50.3	36.1	35.0	15.1	11.5	11.4	23.2	19.9	20.5	18.2
Mistral-Small-24B	46.9	73.9	86.3	76.7	90.8	80.4	73.1	55.5	48.2	48.2	47.2	44.9	40.8	18.2	14.1	13.9	25.6	24.0	24.0	17.7
Gemma-3-27B	57.5	77.0	84.2	78.1	72.9	82.7	61.2	59.5	49.3	50.4	58.6	46.1	45.6	22.7	18.8	18.9	27.6	25.3	25.5	16.3
Grok-4.20	59.8	78.2	90.6	81.0	72.9	82.5	58.8	74.5	54.5	58.6	68.0	35.4	36.6	29.2	14.9	16.5	35.0	33.1	32.9	15.1

Table 9: **Ablation: Automatic evaluation results for the three other prompt-construction settings.** Concl. is the conclusion matching rate. P, R, and F are percentages; GED is an absolute matching-aware graph edit distance ($\downarrow$), not a percentage. Edge scores split into *logos* / *trigger* as in Table 4. The best model for each column and group is **boldfaced**.

Model	Concl.	Element			Premise			Claim			Background			Logos edge			Trigger edge			GED
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
InternVL2.5-26B	10.0	62.3	63.2	62.8	12.5	13.2	12.8	19.6	18.0	18.8	9.1	7.0	7.9	2.7	2.1	2.4	15.9	12.5	14.0	17.2
Qwen2.5-VL-32B	43.3	57.8	65.4	61.4	0.0	0.0	0.0	16.4	14.8	15.5	11.4	11.6	11.5	3.8	3.2	3.5	15.0	11.2	12.9	17.9
Mistral-Small-24B	30.0	65.9	79.4	72.0	36.4	21.1	26.7	30.2	28.7	29.4	24.5	26.7	25.6	10.5	9.5	9.9	27.5	27.5	27.5	15.8
Gemma-3-27B	48.3	69.9	81.3	75.2	21.9	18.4	20.0	45.6	34.7	39.4	43.3	31.0	36.1	20.4	11.8	15.0	18.5	17.1	17.8	13.6
Grok-4.20	53.3	71.6	85.3	77.9	16.7	10.5	12.9	28.8	36.9	32.4	23.0	26.7	24.7	9.4	11.1	10.2	29.4	30.0	29.7	16.4

Table 10: **Human evaluation results for bottom-up generation** ($n = 30$ tasks per model). Concl. is the conclusion match rate; P, R, F are micro-averaged percentages; GED is an absolute matching-aware graph edit distance ($\downarrow$). The best result in each column is **boldfaced**.

↩ 4. Every non-root node must appear as a ↩ premise in at least one argument. --- VALIDATION CHECKLIST (verify before ↩ outputting): - ROOT NODE index never appears in any " ↩ premises" list - Every non-root node appears as a ↩ premise in at least one argument - No cycles exist in the resulting graph - All node indices used in arguments ↩ match indices from the input - Output is valid JSON with an "arguments ↩ " key

I Ablation Studies

Table 9 shows that performance is stable under prompt variation (top-down vs. bottom-up generation) and demonstration selection (most similar vs. random). The model ordering is the same in

all four settings: InternVL well behind, then Qwen and Mistral close together, then Gemma, then Grok. The ordering of aspects by difficulty is likewise stable: logos edges < trigger edges < background < claim < premise < element.

J Human Evaluation Results

To verify the correctness of automatic node matching, we collected human node matching labels on 30 memes sampled from the test set. The prompts in Appendix F were reused as annotation guidelines. Table 10 reports the human evaluation results.

The qualitative picture agrees with the automatic evaluation: element scores are the highest, edges are the hardest, trigger edges are easier than logos edges, and the conclusion match rate is no higher than 53%. The model ranking is broadly preserved—InternVL trails, Mistral is in the middle, and Gemma and Grok lead, trading the top

spot depending on the aspect rather than Grok dominating uniformly. The main discrepancy is Qwen, whose conclusion rate (43%) is high even though its node and edge scores stay low. The premise column rests on only a handful of gold premise nodes in this older annotation set and should be read as indicative only.

We caution against reading too much into the exact human numbers. The per-model coverage is only 30 memes, making the human scores noisier across categories than the automatic ones. This is precisely why the main paper relies on the automatic protocol. We include the human numbers only as a sanity check that the two protocols tell the same story.

K Argument-Graph Annotations for Figure 3

Figure 5 shows the gold argument-graph annotations for the three memes in Figure 3, one per panel. Each node is prefixed with its index and type—**C** (claim), **E** (element), or **B** (background). Indentation encodes support/attack edges directed from a child up to its parent; the root is the meme’s conclusion. A node that supports more than one parent (e.g., [1C] in panel (c)) is listed once in full and referenced thereafter.

```

[0C] Democrats are violent and align themselves with extremist groups across all time
↳ periods
|-- [1C] Democrats were historically aligned with the KKK
|   |-- [2E] [Text] DEMOCRATS THEN
|   |-- [3E] [Image] A large crowd of people wearing white Ku Klux Klan robes and pointed
↳ hoods
|   |-- [10E] [Image] One person in the crowd of white-robed figures has the name "JOHN
↳ PODESTA" printed on their robe
|   |-- [11B] John Podesta is a prominent Democratic political figure who served as the
↳ chairman of Hillary Clinton's 2016 presidential campaign.
|   `-- [12B] Historically, the Democratic Party in the Southern United States was
↳ associated with the Ku Klux Klan during the Reconstruction era.
|-- [4B] the KKK is an extremist group and has a reputation of violence
|-- [5C] Democrats are currently aligned with Antifa
|   |-- [6E] [Text] DEMOCRATS NOW
|   |-- [7E] [Image] A large crowd of people wearing black clothing and face masks
|   `-- [13B] The practice of wearing all-black clothing and face masks at protests is a
↳ tactic commonly associated with the Antifa movement.
`-- [8B] Antifa is an extremist group and has a reputation of violence

```

```

[0C] Joe Biden is racist
|-- [1C] Joe Biden generalizes Latino people based on stereotypes
|   |-- [2E] [Text] "If you don't listen to Despacito, you ain't Latino."
|   |-- [3B] Despacito is a popular song with Spanish lyrics.
|   |-- [4B] Latino people are known to speak Spanish
|   |-- [5E] [Image] Joe Biden with a sombrero and a mustache
|   |-- [7B] The text 'you ain't Latino' parodies Joe Biden's 2020 statement, 'If you have
↳ a problem figuring out whether you're for me or Trump, then you ain't black.'
|   |-- [8B] Joe Biden played the song 'Despacito' on his phone during a 2020 campaign
↳ event for Hispanic voters.
|   `-- [9B] A sombrero and a thick mustache are common visual stereotypes of Mexican or
↳ Latino men.
`-- [6B] Generalizing a race of people based on stereotypes is considered racist

```

```

[0C] Black Lives Matter dedicates effort and resources to actions (like painting streets)
↳ that are ineffective compared to cleaning up blighted communities
|-- [1C] Black Lives Matter puts significant effort into painting streets
|   |-- [2E] [Image] People painting the words "Black Lives Matter" on a street
|   |-- [3B] In June 2020, a two-block-long 'Black Lives Matter' mural was painted on a
↳ street in Washington, D.C. as a gesture of solidarity, and the area was renamed Black
↳ Lives Matter Plaza.
|   `-- [6B] The individuals depicted painting the mural are activists and volunteers
↳ associated with the Black Lives Matter movement.
`-- [4C] Cleaning up blighted communities is a more effective use of effort than painting
↳ streets
|   |-- [5E] [Text] "Imagine the possibilities if they were to put this much effort into
↳ cleaning up blighted communities"
|   |-- [1C] Black Lives Matter puts significant effort into painting streets (repeated
↳ node; see above)
|   `-- [7B] The term 'blighted communities' refers to urban areas experiencing physical
↳ decay, economic disinvestment, and social neglect.

```

Figure 5: Gold argument-graph annotations for the three memes in Figure 3(a)-(c).