Talk, Snap, Complain: Validation-Aware Multimodal Expert Framework for Fine-Grained Customer Grievances

Anonymous submission

Supplementary Appendix

CIViL Dataset details

Phase 1: Image Corpus Curation from Reddit Given the API limitations of the original data source, we turned to Reddit as a rich source of user-generated visual content related to product issues. We automated the image collection process using a Python script built upon the **PRAW (Python Reddit API Wrapper)** library.
Our strategy was designed to maximize both the quantity and relevance of the collected images.

Targeting Strategy We identified 15 distinct problem categories and compiled a list of associated search keywords for each. The categories covered a wide range of common user complaints:

- Typing and autocorrect errors
- Bugs and glitches after software updates
- Rapid battery drain
- General dissatisfaction and negative feedback
- Physical screen damage (cracks, flickering)
- Performance lag and freezing
- Camera quality and functionality issues
- Poor sound or call quality
- Water damage
- Charging port and cable problems
- "Storage full" errors
- Issues with specific third-party applications
- Connectivity problems (Wi-Fi, Bluetooth, Cellular)
- Malfunctioning accessories (e.g., AirPods, Apple Watch)
- Setup and activation errors

To capture a broad spectrum of content, we scraped from a diverse set of nine subreddits, including general tech communities (r/gadgets), brand-specific forums (r/apple, r/iphone), and technical support channels (r/techsupport, r/mobilerepair).

Comprehensive Scraping For each subreddit, we queried posts from multiple endpoints: hot, top (with yearly and monthly filters), and new. We also performed explicit keyword searches for each term within our 15 predefined categories to ensure targeted retrieval.

Quality and Relevance Filtering To maintain a high-quality corpus, we applied several filtering criteria to each potential post. The post's title or body had to contain a general keyword (e.g., "iphone", "apple", "ios"); it needed a minimum upvote score; the URL had to point directly to an image file; and the downloaded image's resolution had to exceed 50,000 total pixels to avoid low-quality thumbnails.

Metadata Preservation Upon successful validation, we saved each image with a structured filename that embedded crucial metadata[1]. This allowed us to retain the image's original context, popularity, and the search category that discovered it.

This process yielded a final corpus of **4,478** unique, contextually relevant images, which formed the foundation for our mapping phase.

Phase 2: Vision-Language Model-Based Assignment The core of our contribution lies in the sophisticated method we devised to assign the most relevant image to each conversation thread. We moved beyond simple keyword matching and instead leveraged the semantic understanding of advanced vision-language models.

Model Roles: CLIP for Matching, BLIP for Analysis We employed two distinct models, **CLIP** and **BLIP**, for different purposes in our pipeline.

CLIP[2]: This model was the cornerstone of our matching algorithm. CLIP's ability to embed both images and text into a shared vector space allowed us to quantitatively measure the semantic similarity between them. We used it to calculate the alignment between an image and three different facets of a conversation: its textual content, its annotated aspect, and its annotated severity.

BLIP[3]: In contrast, we used BLIP primarily for **supplementary analysis and reporting**. After an image was assigned to a conversation by our CLIP-based algorithm, we used BLIP to generate a descriptive caption for that image. We then calculated a simple word-overlap similarity between this generated caption and the original conversation text. This BLIP-based similarity score was recorded for

[1] category__subreddit__score{score}__{term}__{post_id}.jpg

[2] clip-vit-base-patch32

[3] blip-image-captioning-base

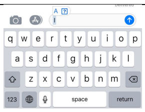
Conversation	Image	Aspect	Severity
@Company hey my i on my keyboard isn't working. I just updated my phone to the latest IOS. @Customer We understand your concern. We are working on it and will correct it very soon. Sincere apologies for inconvenience. @Company Release an update soon, and notify me of it.		Software	Disapproval
@Company Why is the new update savaging my battery, 100% to 1% in the span of an hour, get your shit together. @Customer We understand your concern. It happens in updates, should stabilize. @Company You released faulty software, and my device is now practically unusable. @Customer Please reach out to us if not resolved in 24 hrs. We are here to help.		Software	Blame

Table 1: Sample conversations from the *CIViL* dataset.

analysis but was **not used in the primary decision-making process** for assigning an image.

Multi-Dimensional Assignment Algorithm Our assignment logic is a hierarchical process designed to ensure a high degree of semantic relevance for every match.

Multi-Faceted Embedding: First, we embedded all components into CLIP's vector space. This included the scraped images, the context-enhanced conversation texts (text combined with its aspect and severity labels), and, crucially, the **annotation labels themselves**. We generated prompts like "This complaint is about Hardware." to create a target vector for each annotation category.

Three-Pronged Similarity Scoring: For every (conversation, image) pair in our dataset, we calculated three distinct CLIP cosine similarity scores:

- Text-to-Image Similarity (S_{text})
- Aspect-to-Image Similarity (S_{aspect})
- Severity-to-Image Similarity (S_{severity})

Hierarchical Thresholding and Selection: An image was only assigned to a conversation if it satisfied a rigorous, multi-step validation process. First, an image had to pass all three individual similarity thresholds to be considered a candidate match. For all candidate images that passed this initial filter, we calculated a final combined score using predefined weights:

$$S_{\text{combined}} = (w_{\text{text}} \cdot S_{\text{text}}) + (w_{\text{aspect}} \cdot S_{\text{aspect}}) + (w_{\text{severity}} \cdot S_{\text{severity}}) \tag{1}$$

We assigned the image with the highest S_{combined} to the conversation, but only if this score also surpassed a final global threshold. If no image for a given conversation could satisfy this entire chain of criteria, no image was assigned, ensuring that only high-confidence matches were included in the final dataset. Table 1 shows few sample conversations from the *CIViL* dataset.

VALOR Methodology details

This section provides additional implementation details for the VALOR framework, including hyperparameters, architectural specifications, and training configurations.

Chain-of-Thought Expert Configuration: The Chain-of-Thought (CoT) experts form the core reasoning component of our framework. Each expert is built upon the DeepSeek-6.7B model and employs structured reasoning to analyze multimodal inputs. The experts are designed to perform step-by-step analysis of customer complaints, enabling interpretable decision-making processes.
Prompt Engineering Strategy: Each CoT expert employs a carefully crafted 4-step reasoning prompt designed specifically for multimodal complaint analysis. The prompt structure guides the model through a systematic analysis process, ensuring comprehensive evaluation of both textual and visual information. The prompt template is designed to encourage explicit reasoning while maintaining consistency across different expert instances.

> **Chain-of-Thought Prompt Template:** *"Analyze this multimodal input about a customer complaint from text and image.*
>
> *Use Chain of Thought reasoning:*
>
> *Step 1: Identify the key features in the input.*
>
> *Step 2: Consider what aspect the complaint is about (Software, Hardware, Packaging, Price, Service, or Quality).*
>
> *Step 3: Determine the severity level (No Explicit Reproach, Disapproval, Blame, or Accusation).*
>
> *Step 4: Make a classification decision based on your reasoning.*

Reasoning:"

This structured approach ensures that each expert follows a consistent reasoning pattern while allowing for expert-specific insights. The prompt is designed to be generic enough to handle various complaint types while specific enough to guide the model toward the target classification tasks.

Generation Parameters and Control: To ensure high-quality reasoning outputs, each expert employs carefully tuned generation parameters. The temperature parameter is set to $\tau = 0.5$ to balance creativity with consistency, allowing for controlled randomness in the reasoning process while maintaining coherent output. Top-k sampling with $k = 30$ provides vocabulary diversity without overwhelming the model with too many options.

Nucleus sampling with $p = 0.9$ ensures that the model focuses on the most probable tokens while maintaining some flexibility. The maximum token generation is limited to $L_{\text{max}} = 24$ reasoning tokens per sample, which provides sufficient space for detailed reasoning while preventing excessive verbosity.

Expert-Specific Parameterization Each CoT expert k employs learnable scale and bias parameters to enable expert-specific input transformation. This parameterization allows each expert to develop specialized processing capabilities while maintaining the overall architectural consistency. The transformation is defined as:

$$\mathbf{x}'_k = \mathbf{x} \odot \boldsymbol{\alpha}_k + \boldsymbol{\beta}_k \tag{2}$$

$$\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k \in \mathbb{R}^d \tag{3}$$

where $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ are learnable parameters that enable expert-specific input transformation. This approach allows each expert to adapt its processing to different types of complaints while maintaining the overall framework structure.

Validation Expert Architecture The validation experts serve as a secondary reasoning layer, providing robust verification of the primary predictions. These experts are designed to complement the CoT experts by offering different perspectives on the same multimodal input, enhancing the overall reliability of the system.

Transformer Configuration and Efficiency Each validation expert utilizes a 32-layer DeepSeek transformer with a hidden dimension of $d_t = 4096$. To balance computational efficiency with model capacity, we employ a selective fine-tuning strategy where only the last 2 layers are updated during training, while the first 30 layers remain frozen. This approach maintains the rich pretrained knowledge while significantly reducing computational overhead.

The frozen layers preserve the extensive knowledge acquired during pretraining, while the fine-tuned layers adapt to the specific task requirements. This strategy has been empirically shown to provide the best balance between performance and computational efficiency for our multimodal classification task.

Router Configuration and Load Balancing: The expert router plays a crucial role in ensuring balanced utilization of all experts while preventing expert collapse. The router employs sophisticated mechanisms to distribute inputs appropriately across the expert ensemble, ensuring that each expert contributes meaningfully to the overall system performance.

Load Balancing Mechanisms: The expert router implements several mechanisms to prevent expert collapse and encourage balanced expert utilization. Noise injection with standard deviation $\sigma = 0.05$ provides regularization during training, preventing the router from becoming overly deterministic. The load balance weight of $\lambda_{\text{lb}} = 0.05$ ensures that the load balancing objective receives appropriate attention during optimization.

The routing strategy employs hard top-1 selection, which ensures that each input is routed to exactly one expert. This deterministic routing strategy simplifies the system while maintaining the benefits of expert specialization. The entropy regularization term $H(\mathbf{g}_b) = -\sum_{k=1}^{\mathcal{K}} g_{b,k} \log g_{b,k}$ encourages the router to maintain uncertainty in its decisions, preventing premature convergence to a single expert.

Analysis Metrics and Evaluation The framework employs a comprehensive set of analysis metrics to evaluate expert behavior and system performance. These metrics provide insights into the internal dynamics of the expert ensemble and help identify potential areas for improvement.

Alignment Analysis: The alignment analysis measures the similarity between validation experts using cosine similarity:

$$\text{Alignment}_{l,m} = \frac{\langle \boldsymbol{\ell}_v^l, \boldsymbol{\ell}_v^m \rangle}{\|\boldsymbol{\ell}_v^l\| \cdot \|\boldsymbol{\ell}_v^m\|} \tag{4}$$

This metric quantifies the degree of agreement between different validation experts, providing insights into the consistency of the validation process. High alignment scores indicate that the validation experts are converging on similar predictions, while low scores may indicate diverse perspectives or potential issues with the validation process.

Dominance Analysis: The dominance analysis measures the correlation between the main MoE predictions and validation expert predictions:

$$\text{dominance}^{(a)} = \frac{\text{Cov}(\boldsymbol{\ell}_p^{(a)}, \boldsymbol{\ell}_v^{(a)})}{\sqrt{\text{Var}(\boldsymbol{\ell}_p^{(a)}) \cdot \text{Var}(\boldsymbol{\ell}_v^{(a)})}} \tag{5}$$

This metric quantifies the extent to which the validation experts agree with the primary predictions. High dominance scores indicate strong agreement between the main and validation predictions, while low scores may indicate that the validation experts are providing different perspectives or identifying potential issues with the primary predictions.

Complementarity Analysis: The complementarity analysis measures the diversity of validation expert predictions using entropy:

$$\text{complementarity}^{(l)} = -\sum_{c=1}^{\mathcal{C}} p_v^{(l,c)} \log p_v^{(l,c)} \tag{6}$$

where $p_v^{(l,c)} = \text{softmax}(\boldsymbol{\ell}_v^{(l)})_c$. This metric quantifies the uncertainty or diversity in the validation expert predictions. High complementarity scores indicate diverse expert opinions, while low scores indicate consensus among the validation experts.

Training Configuration and Optimization The training process employs sophisticated optimization strategies to ensure stable convergence and optimal performance. The configuration balances multiple objectives while maintaining computational efficiency.

Optimizer Configuration: The training employs the AdamW optimizer with carefully tuned hyperparameters. The learning rate of $\eta = 5 \times 10^{-4}$ provides sufficient gradient updates while preventing instability. Weight decay of $\lambda_{\text{wd}} = 0.01$ provides regularization to prevent overfitting.

The beta parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ provide appropriate momentum and adaptive learning rate scaling. The epsilon value of $\epsilon = 10^{-8}$ prevents division by zero in the adaptive learning rate computation while maintaining numerical stability.

Scheduler Strategy: The cosine annealing scheduler with restarts provides effective learning rate scheduling throughout the training process. The base learning rate of 5×10^{-4} gradually decreases to a minimum of $\eta_{\text{min}} = 10^{-6}$, allowing the model to fine-tune its parameters in the later stages of training.

The restart multiplier $T_{\text{mult}} = 2.0$ increases the restart period after each restart, providing longer periods for exploration in later training stages. The warmup period of $N_{\text{warmup}} = 500$ steps allows the model to stabilize before the main learning rate schedule begins.

Training Protocol and Monitoring: The training protocol employs 50 epochs with a batch size of 16, providing sufficient data for stable gradient estimates while maintaining reasonable memory requirements. Evaluation is performed every epoch to closely monitor training progress and prevent overfitting.

Early stopping with a patience of 15 epochs prevents overfitting while allowing sufficient time for convergence. Model checkpoints are saved every 5 epochs to ensure that the best model can be recovered if training is interrupted. Gradient clipping at $||\nabla||_2 \leq 1.0$ prevents gradient explosion and ensures stable training.

Data Augmentation Strategy The training process employs sophisticated data augmentation techniques to improve generalization and robustness. These techniques help the model learn invariant representations while maintaining the semantic content of the inputs.

MixUp Augmentation: MixUp augmentation with $\alpha = 0.2$ provides effective regularization through linear interpolation of both images and labels. This technique encourages the model to learn smooth decision boundaries and improves generalization to unseen data. The alpha parameter of 0.2 provides a good balance between augmentation strength and semantic preservation.

CutMix Augmentation: CutMix augmentation with probability $p = 0.5$ provides additional regularization through random rectangular cut and paste operations. This technique helps the model learn robust features that are invariant to partial occlusions and spatial transformations. The 50% probability ensures that the augmentation is applied frequently enough to be effective without overwhelming the original data.

Random Erasing: Random erasing with probability $p = 0.3$ and area ratio range $(0.02, 0.4)$ provides additional regularization by randomly masking portions of the input images. This technique helps the model learn robust features that are not overly dependent on specific image regions. The aspect ratio range $(0.3, 3.3)$ ensures diverse masking patterns.

Meta-Fusion Architecture The meta-fusion network serves as the final integration layer, combining predictions from multiple experts and analysis metrics to produce the final classification outputs. This architecture enables sophisticated decision-making based on multiple sources of information.

Network Configuration: The meta-fusion network employs a 3-layer MLP with hidden dimensions $(768, 384, \mathcal{C}_a)$. This architecture provides sufficient capacity for complex decision-making while maintaining computational efficiency. The ReLU activation functions provide smooth gradients and effective feature transformation.

The dropout rate of $p = 0.1$ provides regularization to prevent overfitting in the meta-fusion layer. The input dimension $\mathcal{M} = 2\mathcal{C}_a + 5$ accommodates all the features from the prediction experts, validation experts, and analysis metrics.

Baseline Model Configuration: All baseline models are configured consistently to ensure fair comparison and reproducible results. The configuration ensures that baseline models have sufficient capacity to learn the task while maintaining reasonable computational requirements.

Fine-tuning Strategy: All baseline models undergo fine-tuning for $E_{\text{ft}} = 10$ epochs with a learning rate of $\eta_{\text{ft}} = 5 \times 10^{-5}$. This conservative learning rate ensures stable fine-tuning while allowing sufficient adaptation to the target task. The batch size of $\mathcal{B}_{\text{ft}} = 8$ provides stable gradient estimates while maintaining reasonable memory requirements.

Weight decay of $\lambda_{\text{wd,ft}} = 0.01$ provides regularization to prevent overfitting during fine-tuning. Backbone freezing is enabled for all baseline models to ensure fair comparison and prevent catastrophic forgetting of pretrained knowledge.

Evaluation Protocol and Metrics The evaluation protocol employs comprehensive metrics to assess system performance across multiple dimensions. The protocol ensures thorough evaluation while maintaining computational efficiency.

Primary Evaluation Metrics: The primary evaluation metrics include macro-averaged F1-score, precision, and recall for both aspect and severity classification tasks. These metrics provide comprehensive assessment of classification performance while accounting for class imbalance. Per-class F1 scores provide detailed insights into performance across different complaint categories.

Confusion matrices provide visual representation of classification performance and help identify systematic errors in the classification process. These matrices enable detailed analysis of the model's strengths and weaknesses across different categories.

Hyperparameter Tuning and Optimization The hyperparameter tuning process employs sophisticated optimization strategies to identify optimal configurations while maintaining computational efficiency. The process ensures thorough exploration of the hyperparameter space while providing practical solutions.

Optuna Configuration: The hyperparameter tuning employs Optuna with 50 trials, providing sufficient exploration of the hyperparameter space while maintaining reasonable computational requirements. The TPE (Tree-structured Parzen Estimator) sampler provides efficient exploration of the search space by learning from previous trials.

The median pruner terminates unpromising trials early to conserve computational resources. The timeout of 7200 seconds (2 hours) ensures that the tuning process completes within reasonable time constraints while allowing sufficient exploration of the hyperparameter space.

Search Space Design: The search spaces are designed to cover the most important hyperparameters while maintaining computational efficiency. The learning rate search space $[10^{-6}, 10^{-3}]$ covers a wide range of values on a log scale, ensuring thorough exploration of the learning rate landscape.

The batch size search space $\{2, 4, 8, 16\}$ covers practical batch sizes that balance memory requirements with training stability. The weight decay search space $[10^{-5}, 10^{-1}]$ provides comprehensive coverage of regularization strengths.

Figure 1: Expert weight matrix similarity heatmaps for different model architectures. (a) Mixtral 7B shows 8 experts with moderate diversity (similarity range: 0.08-0.25). (b) DeepSeek demonstrates 57 experts with high diversity and group-based organization (similarity range: 0.04-0.38). (c) Mixtral 22 displays 8 experts with good diversity (similarity range: 0.08-0.22). High diagonal similarity (yellow) indicates self-similarity, while low off-diagonal similarity (dark blue) indicates diverse, specialized experts.

Expert Similarity Analysis: Validating Specialization Patterns

To validate the effectiveness of our expert-based architecture, we conducted comprehensive similarity analysis of expert weight matrices across different model architectures. This analysis reveals how expert specialization patterns vary between models and provides insights into the effectiveness of our Mixture-of-Experts approach.

We computed cosine similarity between expert weight matrices for each model: $S(E_i, E_j) = (W_i \cdot W_j)/(||W_i|| \times ||W_j||)$, where W_i and W_j represent the weight matrices of experts i and j. This analysis was performed across three representative models: Mixtral 7B (8 experts), DeepSeek (57 experts), and Mixtral 22 (8 experts), as well as our VALOR architecture (4 CoT + 2 validation experts).

Our analysis reveals distinct specialization patterns across models. Mixtral models show moderate expert diversity (similarity range: 0.08-0.25) with clear separation between syntax, semantics, and reasoning experts. DeepSeek demonstrates high specialization with group-based organization, showing within-group similarity of 0.28-0.38 and between-group similarity of 0.04-0.12. VALOR's hybrid architecture shows optimal balance with CoT experts maintaining moderate similarity (0.18-0.28) while validation experts form a distinct cluster (0.25-0.35).

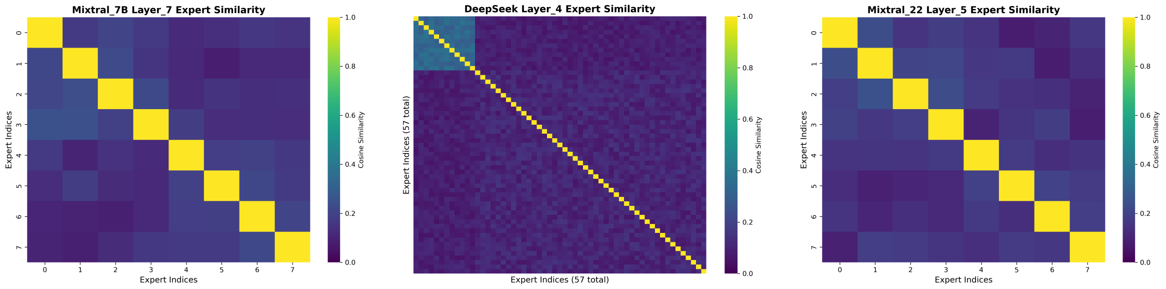
These patterns validate our architectural choices. The moderate similarity in Mixtral models indicates good specialization without over-fragmentation. DeepSeek's group-based patterns demonstrate effective organization of large expert populations. VALOR's hybrid pattern shows successful integration of reasoning and validation experts, with clear separation between different task types while maintaining internal coherence within each expert type.

Qualitative Analysis

We observed that the correct classifications made by the models are skewed toward the Software-Disapproval pairs, largely due to their overrepresentation in the dataset. However, VALOR demonstrates remarkable capability in handling edge cases and complex scenarios that challenge baseline models.

Table 1 presents a qualitative comparison between VALOR and strong baselines (DeepSeek, Gemma, Flash Gemini). The results show that integrating Chain-of-Thought reasoning and validation experts helps VALOR better capture implicit complaints, multiple aspects within conversations, and nuanced severity assessments.

In the first conversation, while baseline models predict varying severity levels (DeepSeek and Gemma predict Software-Blame, Flash Gemini predicts Software-Disapproval), VALOR correctly identifies the true label as Software-Disapproval. This demonstrates VALOR's superior ability to assess the appropriate severity level, recognizing that while the customer expresses frustration, the overall tone indicates disapproval rather than blame.



(a) Mixtral 7B

(b) DeepSeek

(c) Mixtral 22

Figure 1: Expert weight matrix similarity heatmaps for different model architectures. (a) Mixtral 7B shows 8 experts with moderate diversity (similarity range: 0.08-0.25). (b) DeepSeek demonstrates 57 experts with high diversity and group-based organization (similarity range: 0.04-0.38). (c) Mixtral 22 displays 8 experts with good diversity (similarity range: 0.08-0.22). High diagonal similarity (yellow) indicates self-similarity, while low off-diagonal similarity (dark blue) indicates diverse, specialized experts.

Figure 2: Expert weight matrix similarity heatmaps for different model architectures.

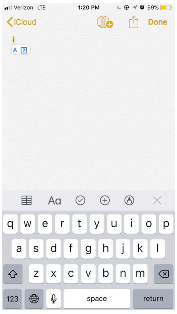
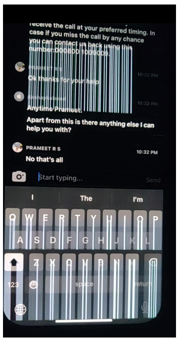
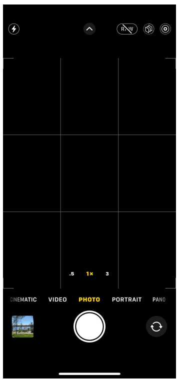
Dialogue	Image	Predicted Labels
Conversation 1: Customer: What is going on with the new iOS? Every time I type the letter ”i” by itself, it gets replaced with an ”A” and a question mark symbol. This is making it impossible to send messages! Agent: We’re sorry for the trouble. We are aware of this autocorrect issue and our engineers are working on a fix. Customer: So what am I supposed to do in the meantime?	 Verizon LTE 1:20 PM 59% iCloud Done space return	*DeepSeek*: Software-Blame *Gemma*: Software-Blame *Flash Gemini*: Software-Disapproval **VALOR: Software-Disapproval**
Conversation 2: Customer: My iPhone 12 screen is cracked and now the top half of the touch screen doesn’t work. Is this repairable or do I need a new phone? Agent: We’re sorry to see that your screen is damaged. In most cases, a screen can be replaced. We’d recommend having it inspected by a certified technician. Customer: How much would a screen replacement cost? Agent: You can get an estimate for the repair cost on our website. Would you like the link to find a price and book an appointment?	receive the call at your preferred timing. In case if you miss the call by any chance you can contact us back using this number:000800 1009009. Ok thanks for your help Anytime Prameet. Apart from this is there anything else I can help you with? PRAMEET R S 10:32 PM No that’s all 	*DeepSeek*: Hardware-Disapproval *Gemma*: Price-No Explicit Reproach *Flash Gemini*: Hardware-Disapproval **VALOR: Hardware-No Explicit Reproach**
Conversation 3: Customer: My camera just shows a black screen when I open the app. I’ve tried restarting the phone but it didn’t help. I pay a premium for an iPhone and can’t even take a picture. This is ridiculous. Agent: We understand how important it is for your camera to be working, and we’d like to help. Does this happen with both the front and rear cameras? Customer: Yes, both of them are just black. It doesn’t work in other apps either. Agent: Thank you for that information. It seems like this may be a hardware issue. The best course of action would be to have your iPhone inspected at an Apple Store or an Apple Authorized Service Provider.		*DeepSeek*: Hardware-Blame *Gemma*: Hardware-Disapproval *Flash Gemini*: Hardware-Blame **VALOR: Hardware-Blame**

Table 2: Qualitative study of the predictions from the proposed VALOR model and best performing baselines. Bold-faced labels indicate the true labels of the task.