# CAP: Conformalized Abstention Policies for Context-Adaptive Risk Management for LLMs and VLMs

**Sina Tayebati** STAYEB3@UIC.EDU
**Divake Kumar** DKUMAR33@UIC.EDU
**Nastaran Darabi** NDARAB2@UIC.EDU
**Dinithi Jayasuriya** DKASTH2@UIC.EDU
**Theja Tulabandhula** THEJA@UIC.EDU
*University of Illinois Chicago, Illinois, United States*
**Ranganath Krishnan*** RANGANATH.KRISHNAN@CAPITALONE.COM
*AI Labs, Capital One, Texas, United States*
**Amit Ranjan Trivedi** AMITRT@UIC.EDU
*University of Illinois Chicago, Illinois, United States*

**Editors:** Hung-yi Lee and Tongliang Liu

# Supplementary Details

## A. Training Procedure

Algorithm 1 summarizes the training of our proposed adaptive conformal environment and abstention policy. In each episode, the policy network samples candidate confidence levels $(\alpha, \beta)$, converts them into *predict* and *abstain* quantiles by computing nonconformity scores on the calibration set, and applies these thresholds to each test input to stochastically select among single-label prediction, set prediction, or abstention using sigmoid-smoothed gates. An episode-level cost function that penalizes misclassification, excessive prediction set size, and unnecessary abstention is evaluated; its negative serves as the reward to update the policy via REINFORCE. Through repeated interactions, the agent learns optimal quantile thresholds $\hat{q}_{\text{predict}}, \hat{q}_{\text{abstain}}$ that minimize expected cost while preserving conformal validity.

## B. Models and Datasets used in this Study

We evaluate six foundation models: three VLMs (LLaVA-v1.6 at 7B, 13B, 34B) and three LLMs (Yi-34B, Qwen-14B, Qwen-7B). This selection spans 7B–34B parameters and includes both unimodal and multimodal architectures. The key characteristics of these models are shown in the table below.

Ten MCQA benchmarks are used—five VLM (MMBench, OODCV-VQA, ScienceQA, SEED-Bench, AI2D) and five LLM (MMLU, CosmosQA, HellaSwag, HalluDial, HaluSum). All are reformatted to fixed-choice format to standardize risk estimation and abstention

* Work done while at Intel Labs

**Algorithm 1** Conformalized Abstention Policy with Reinforcement Learning

**Input:** Calibration dataset $\mathcal{D}_{\text{cal}}$, LLM/VLM model $M$, learning rate $\eta$, policy network $\pi_\theta$, cost function $C(\alpha, \beta)$
**Output:** Optimized thresholds $\hat{q}_{\text{predict}}$, $\hat{q}_{\text{abstain}}$ **for** *each episode* **do**
  **end**
  Sample $\alpha \sim \mathcal{N}(\mu_\theta^{(\alpha)}, \sigma_\theta^{(\alpha)2})$ and $\beta \sim \mathcal{N}(\mu_\theta^{(\beta)}, \sigma_\theta^{(\beta)2})$
Compute nonconformity scores $s_i = 1 - p_{y_i}(\mathbf{x}_i)$ for all $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$
Calculate quantile thresholds:
  $\hat{q}_{\text{predict}} = \text{Quantile}(\{s_i\}, (n+1)(1-\alpha)/n)$
  $\hat{q}_{\text{abstain}} = \text{Quantile}(\{s_i\}, (n+1)(1-\beta)/n)$ **for** *each test sample* $\mathbf{x}$ **do**
  **end**
  Compute $s(\mathbf{x}) = 1 - \max_i p_i(\mathbf{x})$
Compute action probabilities:
  $p_{\text{single}} = \sigma(-c[s(\mathbf{x}) - \hat{q}_{\text{predict}}])$
  $p_{\text{abstain}} = \sigma(c[s(\mathbf{x}) - \hat{q}_{\text{abstain}}])$
  $p_{\text{set}} = 1 - p_{\text{single}} - p_{\text{abstain}}$
Sample action $a \in \{\text{single}, \text{set}, \text{abstain}\}$ based on these probabilities

Evaluate performance and compute cost $C(\alpha, \beta)$
Compute reward $R(\alpha, \beta) = -C(\alpha, \beta)$
Update policy parameters:
  $\theta \leftarrow \theta + \eta \cdot R(\alpha, \beta) \nabla_\theta \log \pi_\theta(\alpha, \beta)$

Table 1: LLM and VLM models used in evaluation.

| Model | Type / Size | Context | Notes |
|---|---|---|---|
| LLaVA-v1.6-34B | VLM / 34B | 8K | Vicuna backbone, visual input |
| LLaVA-v1.6-13B | VLM / 13B | 8K | Multimodal reasoning |
| LLaVA-v1.6-7B | VLM / 7B | 8K | Efficient visual–text fusion |
| Yi-34B | LLM / 34B | 200K | Bilingual, trained on 3T tokens |
| Qwen-14B | LLM / 14B | 32K | Multilingual, strong reasoning |
| Qwen-7B | LLM / 7B | 8K–32K | Lightweight, 2.4T training tokens |

behavior. Each dataset is split 50/50 into calibration and test sets. This ensures sufficient data for both conformal thresholding and evaluation.

## C. Prompting Templates

This section describes the prompting strategies and templates used for evaluating VLMs and LLMs. The templates are designed to ensure consistent and effective evaluation across different model families.

Table 2: MCQA benchmarks used in this study. Counts are rounded for brevity.

| Dataset | Modality | Key characteristics |
|---|---|---|
| MMBench | V + L | 3 000+ MC questions spanning 20 ability dimensions; CircularEval scoring. |
| OODCV-VQA (Digits) | V + L | Out-of-distribution digit-counting subset with 2-choice questions. |
| ScienceQA | V + L | 21 208 multimodal science questions with image/text context, lectures and explanations. |
| SEED-Bench | V + L | 24 k MC questions across 27 capability dimensions evaluating text & image generation. |
| AI2D | V + L | 5 k science diagrams, 150 k annotations, 15 k diagram-based MC questions. |
| MMLU | Text | 57 subject-area tasks (elementary → professional) testing broad knowledge. |
| CosmosQA | Text | 35.6 k commonsense narrative comprehension problems. |
| HellaSwag | Text | 70 k adversarially-filtered grounded commonsense inference questions. |
| HalluDial | Text | 4 094 dialogues (146 k samples) for dialogue-level hallucination detection. |
| HaluSum | Text | Summarization subset of the 35 k HaluEval benchmark targeting hallucinations. |

**Prompting Templates for VLMs:** For Vision-Language Models, we adopt a standardized prompting strategy tailored for multiple-choice visual question answering (MCQA) tasks. The template is inspired by the approach used in LLaVA and is designed to maximize compatibility across various VLM architectures. The prompt structure is as follows:

- The prompt begins with an attached image, serving as the primary visual input for the model.
- The question text follows, optionally including a hint if available.
- Six answer options are presented line by line, each prefixed with its corresponding letter (A-F). Additional choices such as *"I don't know"* and *"None of the above"* are also included to account for uncertainty.
- The prompt concludes with an explicit instruction: *"Answer with the option's letter from the given choices directly."*
- For models requiring a specific multimodal token format, the image is prepended with a designated image token, such as <image> or model-specific tokens like `DEFAULT_IMAGE_TOKEN`, ensuring compatibility with different VLM architectures.

- Depending on the model type, the prompt is wrapped within a structured conversational template. Examples include Vicuna-style conversation for LLaVA, structured input for CogVLM, Yi-VL, and Qwen-VL, ensuring consistency in processing.

To accommodate the constraints of single-image input in many VLMs, we intentionally exclude few-shot demonstrations from the prompts. The templates are adapted for specific model families, including LLaVA, Yi-VL, Qwen, Monkey, MoE-LLaVA, mPLUG-Owl, and MobileVLM, using their respective official repositories. For CogAgent and InternLM-XComposer2, the templates are sourced from their Hugging Face repositories.

Below is the base prompt template format utilized in our experiments:

```
Image: {<Image>}

Question: {Question Text}

Hint: {Optional Hint Text}

Choices:

A. {Content of option A}
B. {Content of option B}
C. {Content of option C}
D. {Content of option D}
E. I don't know
F. None of the above

Answer with the option's letter from the given choices directly.
```

Table 3: This table presents the structured prompt template used for multiple-choice question answering in VLMs. Each prompt consists of an attached image, a question (optionally with a hint), and six answer choices, including uncertainty options (*"I don't know"* and *"None of the above"*). To maintain consistency across different VLM architectures, model-specific input tokens (e.g., <image> or DEFAULT_IMAGE_TOKEN) are included when necessary. The prompt concludes with a direct instruction for the model to answer using the letter corresponding to its chosen option.

This template ensures a consistent format for evaluating VLMs across diverse datasets and tasks. The inclusion of six options (A-F) standardizes the evaluation process, while the explicit instruction at the end guides the model to provide a direct response.

**Prompting Templates for LLMs:** For Language Models, we employ a **base prompting strategy** without any strategy such as shared instruction or task-specific instruction prompt in order maintain a standardized approach across evaluations. This prompt is designed to evaluate several model performances across multiple tasks, including question answering (QA), reading comprehension (RC), commonsense inference (CI), dialogue response selection (DRS), and document summarization (DS). The prompt template is designed to provide a consistent structure for all tasks while accommodating task-specific information. The structure of the base prompt is as follows:

- The prompt begins with the task-specific context, dialogue, or document:

  - For *QA tasks*, no background information is included.
  - For *RC and CI tasks*, the keyword *"Context"* introduces the relevant background information.
  - For *DRS tasks*, the keyword *"Dialogue"* incorporates the dialogue history.
  - For *DS tasks*, the keyword *"Document"* includes the document content.
- The question is presented next, followed by a list of six answer options:
  - Four standard options (A-D) with task-specific content.
  - Two additional options: *"I don't know"* and *"None of the above."*
- The model is instructed to provide the letter corresponding to the correct answer.

Below is the base prompt template format utilized in our experiments:

```
Context/Dialogue/Document: {The context or dialogue history or document
corresponding to the following question}

Question: {Question}

Choices:

A. {Content of option A}
B. {Content of option B}
C. {Content of option C}
D. {Content of option D}
E. I don't know
F. None of the above

Answer with the option's letter from the given choices directly.
```

Table 4: This table presents the structured prompt template used for multiple-choice question answering in LLMs. In the QA setting, no additional background information is included. For the RC and CI tasks, the keyword "Context" is introduced to incorporate relevant background information. Similarly, the keywords "Dialogue" and "Document" are used for DRS and DS tasks, respectively, to integrate necessary context.

This template ensures a standardized format for evaluating LLMs across diverse tasks. For instruction-finetuned LLMs, the entire prompt input is treated as the user's message, and the "apply_chat_template" function is used to transform the prompt into a chat format, ensuring compatibility with chat-based models.

## D. Additional Results

We provide additional details on the Vision-Language Models (VLMs) and Large Language Models (LLMs), complementing the main body of the paper. We evaluated these models to broaden our analysis across various architectures and parameter scales.

For **VLMs**, we include results for several additional models: **Monkey-Chat 7B** (optimized for multimodal chat-based reasoning), **InternLM-XComposer2-VL 7B** (enhances

vision-language interaction through structured prompts), **Yi-VL 6B** (a smaller variant of the Yi-VL series, designed for effective image-text understanding), **CogAgent-VQA 7B** (focuses on visual question answering with robust reasoning capabilities), **MobileVLMV2 7B** (a lightweight VLM tailored for mobile and edge applications), **mPLUG-Owl2 7B** (offers strong image-text understanding capabilities), and **Qwen-VL-Chat 7B** (designed for dialogue-driven multimodal interactions). For **LLMs**, we also present results for the **Llama-2 7B and 13B** models, which serve as foundation models with strong text generation and reasoning capabilities.

### D.1. Results of VLMs

Additional results in Tables 5, 6, and 7 demonstrate the performance of multiple VLMs in terms of uncertainty quantification: AUROC vs. AUARC, coverage rate vs. set size, and accuracy vs. expected calibration error, respectively. As shown in these tables, our CAP model outperforms other methods in hallucination detection and uncertainty-guided selective generation while satisfying a minimum coverage rate of 90% and maintaining a balanced set size across all cases.

Table 5: Evaluation of uncertainty quantification: Comparative analysis of the proposed CAP (Ours) method with standard Least Ambiguous Set-valued Classifiers (LAC) and Adaptive Prediction Sets (APS) methods (best values are in bold). The comparison includes different datasets and VLM models, with quality of uncertainty quantification evaluated using the Area Under the Receiver Operating Characteristic (AUROC) and the Area Under the Accuracy-Rejection Curve (AUARC).

| **Model** | **Method** | **AUROC** ↑ (Hallucination Detection) | | | | | | **AUARC** ↑ (Uncertainty guided selective generation) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **MMB** | **OOD** | **SQA** | **SB** | **AI2D** | **Avg.** | **MMB** | **OOD** | **SQA** | **SB** | **AI2D** | **Avg.** |
| Monkey-Chat-7B | APS | 0.6360 | 0.2994 | 0.4916 | 0.5304 | 0.7662 | 0.5447 | 0.9285 | 0.7640 | 0.8950 | 0.8579 | 0.8635 | 0.8618 |
| | LAC | 0.6855 | 0.4151 | 0.6501 | 0.4596 | 0.6716 | 0.5764 | 0.8988 | 0.7137 | 0.8646 | 0.8028 | 0.8413 | 0.8242 |
| | Ours | **0.7241** | **0.5182** | **0.6739** | **0.5550** | **0.7340** | **0.6410** | **0.9652** | **0.9174** | **0.9686** | **0.9335** | **0.9747** | **0.9519** |
| InternLM-XComposer2-VL-7B | APS | 0.6648 | 0.5000 | 0.7010 | 0.4731 | 0.6421 | 0.5962 | 0.9267 | 0.7999 | 0.9537 | 0.8642 | 0.8879 | 0.8865 |
| | LAC | 0.6861 | 0.5275 | 0.7524 | 0.4810 | 0.6429 | 0.6180 | 0.9001 | 0.7807 | 0.9301 | 0.8322 | 0.8624 | 0.8611 |
| | Ours | **0.7068** | **0.6295** | **0.7909** | **0.5773** | **0.7035** | **0.6816** | **0.9667** | **0.9219** | **0.9762** | **0.9261** | **0.9624** | **0.9507** |
| CogAgent-VQA-7B | APS | 0.6416 | 0.3448 | 0.4930 | 0.5274 | **0.5341** | 0.5082 | 0.9240 | 0.7469 | 0.8448 | 0.8741 | 0.7828 | 0.8345 |
| | LAC | 0.7003 | 0.3396 | 0.5693 | 0.4844 | 0.4245 | 0.5036 | 0.8996 | 0.7015 | 0.8130 | 0.8251 | 0.7483 | 0.7975 |
| | Ours | **0.7432** | **0.5175** | **0.6355** | **0.5346** | 0.4867 | **0.5835** | **0.9746** | **0.9264** | **0.9608** | **0.9471** | **0.9553** | **0.9528** |
| MobileVLM-v2-7B | APS | **0.7646** | 0.3836 | 0.5652 | 0.4153 | 0.4867 | 0.5231 | 0.9610 | 0.8712 | 0.9503 | 0.9296 | 0.8508 | 0.9126 |
| | LAC | 0.7168 | 0.3963 | 0.6777 | 0.4617 | 0.3539 | 0.5213 | 0.9307 | 0.8196 | 0.9133 | 0.8673 | 0.7866 | 0.8635 |
| | Ours | 0.7368 | **0.5214** | **0.6672** | **0.5695** | **0.4633** | **0.5916** | **0.9682** | **0.9169** | **0.9698** | **0.9194** | **0.9103** | **0.9369** |
| mPLUG-Owl2-7B | APS | 0.5347 | 0.4550 | 0.3855 | 0.3421 | 0.4862 | 0.4407 | 0.9625 | 0.8706 | 0.9111 | 0.9134 | 0.8628 | 0.9041 |
| | LAC | 0.6575 | 0.5069 | 0.4828 | 0.3692 | 0.3432 | 0.4719 | 0.9247 | 0.8383 | 0.8677 | 0.8447 | 0.7949 | 0.8541 |
| | Ours | **0.6920** | **0.6316** | **0.5766** | **0.5169** | **0.4792** | **0.5793** | **0.9650** | **0.9244** | **0.9415** | **0.9051** | **0.9066** | **0.9285** |
| Qwen-VL-Chat-7B | APS | 0.6230 | 0.4610 | 0.5156 | 0.4990 | 0.6786 | 0.5554 | 0.8882 | 0.6872 | 0.8052 | 0.7918 | 0.8536 | 0.8052 |
| | LAC | 0.6557 | 0.4057 | 0.5394 | 0.4624 | 0.6511 | 0.5429 | 0.8593 | 0.6425 | 0.7851 | 0.7616 | 0.8292 | 0.7755 |
| | Ours | **0.6907** | **0.5348** | **0.6079** | **0.5481** | **0.6990** | **0.6161** | **0.9600** | **0.9171** | **0.9313** | **0.9262** | **0.9688** | **0.9407** |
| Yi-VL-6B | APS | 0.6094 | 0.3616 | 0.5674 | 0.4747 | 0.4486 | 0.4923 | 0.9517 | 0.8790 | 0.9012 | 0.9023 | 0.8747 | 0.9018 |
| | LAC | 0.6785 | 0.4638 | 0.5780 | 0.4387 | 0.4246 | 0.5167 | 0.9198 | 0.8461 | 0.8606 | 0.8501 | 0.8276 | 0.8608 |
| | Ours | **0.7432** | **0.6284** | **0.6446** | **0.5471** | **0.5331** | **0.6193** | **0.9676** | **0.9228** | **0.9551** | **0.9187** | **0.9312** | **0.9391** |
| MoE-LLaVA-Phi2-2.7B | APS | 0.6359 | 0.5785 | 0.5248 | 0.4199 | 0.4282 | 0.5175 | 0.9446 | 0.7610 | 0.8522 | 0.8815 | 0.8061 | 0.8491 |
| | LAC | 0.6864 | 0.5614 | 0.4810 | 0.4849 | 0.4142 | 0.5256 | 0.9070 | 0.7360 | 0.8083 | 0.8298 | 0.7576 | 0.8077 |
| | Ours | **0.7360** | **0.7147** | **0.5329** | **0.5772** | **0.5352** | **0.6192** | **0.9655** | **0.9477** | **0.9412** | **0.9342** | **0.9284** | **0.9434** |

Table 6: Evaluation of coverage rate (%) and set size: Comparative analysis of the proposed CAP (Ours) method with standard LAC and APS methods. The comparison includes different datasets and VLM models, showcasing the satisfied coverage rate and balanced set sizes produced by our method with underlined values.

| Model | Method | Coverage (%) ↑ | | | | | | SS ↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MMB | OOD | SQA | SB | AI2D | Avg. | MMB | OOD | SQA | SB | AI2D | Avg. |
| Monkey-Chat-7B | APS | 97.85 | 96.27 | 98.84 | 96.50 | 97.28 | 97.35 | 3.787 | 3.669 | 3.455 | 4.013 | 4.040 | 3.793 |
| | LAC | 89.45 | 88.75 | 90.44 | 89.22 | 90.98 | 89.77 | 1.611 | 2.181 | 1.656 | 2.505 | 2.346 | 2.060 |
| | Ours | 93.33 | 91.35 | 94.69 | 92.03 | 94.36 | 93.15 | 2.383 | 2.987 | 2.567 | 3.285 | 3.017 | 2.848 |
| InternLM-XComposer2-VL-7B | APS | 96.57 | 92.48 | 98.74 | 94.46 | 96.28 | 95.71 | 3.479 | 2.575 | 3.383 | 3.578 | 3.673 | 3.338 |
| | LAC | 89.17 | 88.96 | 89.58 | 89.90 | 89.87 | 89.50 | 1.966 | 1.819 | 1.443 | 2.584 | 2.358 | 2.034 |
| | Ours | 93.51 | 90.01 | 92.97 | 90.21 | 92.43 | 91.82 | 2.763 | 2.457 | 1.926 | 3.123 | 2.902 | 2.634 |
| CogAgent-VQA-7B | APS | 98.54 | 95.64 | 97.47 | 95.94 | 93.83 | 96.28 | 2.997 | 2.944 | 2.833 | 2.996 | 3.240 | 3.002 |
| | LAC | 90.68 | 90.37 | 90.14 | 89.36 | 90.65 | 90.24 | 1.665 | 1.971 | 1.895 | 1.975 | 2.640 | 2.030 |
| | Ours | 94.15 | 92.12 | 93.53 | 93.59 | 94.25 | 93.53 | 2.175 | 2.757 | 2.506 | 3.015 | 3.652 | 2.821 |
| MobileVLM-v2-7B | APS | 97.99 | 96.27 | 99.04 | 97.67 | 95.87 | 97.37 | 3.439 | 3.074 | 3.610 | 3.494 | 3.866 | 3.497 |
| | LAC | 89.63 | 90.86 | 89.07 | 89.49 | 90.23 | 89.86 | 1.629 | 2.153 | 1.625 | 2.106 | 2.925 | 2.088 |
| | Ours | 92.78 | 91.49 | 94.18 | 91.53 | 90.23 | 92.04 | 2.159 | 2.623 | 2.329 | 2.567 | 3.448 | 2.625 |
| mPLUG-Owl2-7B | APS | 99.27 | 95.08 | 98.18 | 97.09 | 95.81 | 97.09 | 3.365 | 2.485 | 3.346 | 3.431 | 3.379 | 3.201 |
| | LAC | 89.81 | 89.52 | 91.40 | 89.94 | 90.34 | 90.20 | 1.727 | 1.689 | 2.070 | 2.432 | 2.624 | 2.109 |
| | Ours | 92.65 | 91.28 | 91.91 | 91.94 | 89.52 | 91.46 | 2.080 | 2.062 | 2.401 | 2.753 | 2.934 | 2.446 |
| Qwen-VL-Chat-7B | APS | 96.21 | 93.46 | 92.01 | 92.97 | 96.70 | 94.27 | 3.413 | 3.589 | 3.349 | 3.692 | 3.796 | 3.568 |
| | LAC | 88.44 | 88.75 | 88.11 | 89.21 | 89.69 | 88.84 | 1.990 | 3.049 | 2.451 | 2.945 | 2.394 | 2.566 |
| | Ours | 93.01 | 90.37 | 90.44 | 91.06 | 93.94 | 91.76 | 2.665 | 3.673 | 3.074 | 3.504 | 3.061 | 3.195 |
| Yi-VL-6B | APS | 98.63 | 95.43 | 98.13 | 95.94 | 96.77 | 96.98 | 3.326 | 2.506 | 3.503 | 3.116 | 3.491 | 3.189 |
| | LAC | 90.22 | 89.94 | 89.78 | 89.84 | 91.01 | 90.16 | 1.621 | 1.536 | 2.009 | 2.106 | 2.514 | 1.957 |
| | Ours | 93.38 | 91.35 | 92.41 | 90.11 | 90.99 | 91.61 | 2.082 | 1.962 | 2.574 | 2.522 | 2.927 | 2.414 |
| MoE-LLaVA-Phi2-2.7B | APS | 99.50 | 93.95 | 97.07 | 97.60 | 96.50 | 96.92 | 3.4961 | 2.2651 | 3.2969 | 3.3834 | 3.3425 | 3.1568 |
| | LAC | 89.26 | 89.17 | 90.84 | 89.66 | 90.08 | 89.80 | 1.5843 | 1.5204 | 2.0976 | 2.0021 | 2.4891 | 1.9387 |
| | Ours | 92.10 | 91.63 | 92.67 | 91.34 | 90.84 | 91.72 | 2.0461 | 2.1280 | 2.6631 | 2.5178 | 2.9515 | 2.4613 |

### D.2. Results of LLMs

Additional results in Tables 8, 9, and 10 demonstrate the performance of Llama-2 series models (7B and 13B) in terms of uncertainty quantification: AUROC vs. AUARC, coverage rate vs. set size, and accuracy vs. expected calibration error, respectively. As shown in these tables, our CAP model outperforms other methods in hallucination detection and uncertainty-guided selective generation while satisfying a minimum coverage rate of 90% and maintaining a balanced set size across all cases.

### D.3. Accuracy vs. ECE

Figure 2 shows the accuracy vs. ECE for CAP, APS, and LAC across multiple VLMs. Lower ECE values indicate better calibration, meaning confidence scores are more reliable indicators of prediction accuracy. As shown, CAP improves accuracy while significantly reducing the expected calibration error. Figure 4 shows the same trend for LLMs, consistently reducing ECE while improving accuracy across all tasks and datasets.

### D.4. Effect of Model Scale

We analyzed the impact of model scale on our CAP method's performance across models of varying sizes. As shown in Figure 5, larger models generally achieve higher accuracy, with the most significant gains observed when scaling from 13B to 34B parameters. Prediction

Table 7: Evaluation of accuracy (%) and ECE: Comparative analysis of the proposed CAP (Ours) method with standard LAC and APS methods. The comparison includes different datasets and VLM models, demonstrating a significant reduction in expected calibration error while improving overall accuracy.

| Model | Method | Accuracy (%) ↑ | | | | | | ECE ↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MMB | OOD | SQA | SB | AI2D | Avg. | MMB | OOD | SQA | SB | AI2D | Avg. |
| Monkey-Chat-7B | APS | 81.40 | 76.75 | 79.27 | 72.33 | 73.58 | 76.67 | 0.2134 | 0.3583 | 0.2696 | 0.2825 | 0.3237 | 0.2895 |
| | LAC | 81.26 | 77.06 | 80.89 | 72.58 | 74.53 | 77.26 | 0.1480 | 0.3042 | 0.1857 | 0.2494 | 0.2608 | 0.2296 |
| | Ours | **84.03** | **78.61** | **82.37** | **74.57** | **78.70** | **79.66** | **0.0159** | **0.0336** | **0.0190** | **0.0336** | **0.0207** | **0.0246** |
| InternLM-XComposer2-VL-7B | APS | 76.72 | 77.04 | 81.73 | 70.80 | 72.52 | 75.76 | 0.1805 | 0.2179 | 0.1727 | 0.2203 | 0.2417 | 0.2066 |
| | LAC | 77.30 | 77.88 | 82.72 | 71.72 | 73.13 | 76.55 | 0.1284 | 0.1871 | 0.1073 | 0.2093 | 0.1776 | 0.1620 |
| | Ours | **78.46** | **78.07** | **84.10** | **72.15** | **75.02** | **77.56** | **0.0341** | **0.0173** | **0.0246** | **0.0593** | **0.0289** | **0.0328** |
| CogAgent-VQA-7B | APS | 81.07 | 76.86 | 75.98 | 76.03 | 67.99 | 75.59 | 0.2310 | 0.3614 | 0.3043 | 0.2327 | 0.3148 | 0.2889 |
| | LAC | 80.55 | 76.91 | 76.16 | 75.48 | 67.98 | 75.42 | 0.1608 | 0.3407 | 0.2340 | 0.2151 | 0.2912 | 0.2484 |
| | Ours | **83.29** | **79.72** | **79.25** | **76.18** | **69.60** | **77.61** | **0.0134** | **0.0470** | **0.0366** | **0.0246** | **0.0110** | **0.0265** |
| MobileVLM-v2-7B | APS | 80.79 | 74.86 | 78.11 | **74.30** | 63.37 | 74.29 | 0.1460 | 0.2230 | 0.1790 | 0.1691 | 0.2838 | 0.2002 |
| | LAC | 80.85 | 75.23 | 78.90 | 74.28 | 63.41 | 74.53 | 0.1202 | 0.2239 | 0.1363 | 0.1927 | 0.2932 | 0.1933 |
| | Ours | **82.12** | **75.38** | **79.66** | 73.95 | **64.77** | **75.18** | **0.0464** | **0.0503** | **0.0306** | **0.0780** | **0.0643** | **0.0539** |
| mPLUG-Owl2-7B | APS | 78.94 | 80.48 | 73.87 | **70.54** | 65.42 | 73.85 | 0.1578 | 0.1858 | 0.2260 | 0.1982 | 0.2588 | 0.2053 |
| | LAC | 78.67 | 79.91 | 74.53 | 69.76 | 64.91 | 73.55 | 0.1453 | 0.1574 | 0.2093 | 0.2384 | 0.2857 | 0.2072 |
| | Ours | **79.78** | **80.88** | **75.28** | 69.88 | **65.96** | **74.36** | **0.0473** | **0.0352** | **0.0439** | **0.0863** | **0.0668** | **0.0559** |
| Qwen-VL-Chat-7B | APS | 76.71 | 62.45 | 70.04 | 66.86 | 71.25 | 69.46 | 0.2350 | 0.3944 | 0.2487 | 0.2937 | 0.3211 | 0.2986 |
| | LAC | 76.78 | 63.91 | 71.12 | 67.42 | 72.10 | 70.27 | 0.1653 | 0.3684 | 0.2249 | 0.2739 | 0.2510 | 0.2567 |
| | Ours | **79.70** | **67.14** | **73.18** | **69.89** | **76.11** | **73.20** | **0.0167** | **0.0316** | **0.0403** | **0.0379** | **0.0087** | **0.0270** |
| Yi-VL-6B | APS | 80.54 | **81.23** | 74.40 | **74.59** | 67.98 | 75.75 | 0.1694 | 0.1720 | 0.2553 | 0.1837 | 0.2621 | 0.2085 |
| | LAC | 80.70 | 80.67 | 74.77 | 74.18 | 68.29 | 75.72 | 0.1263 | 0.1594 | 0.2136 | 0.2019 | 0.2565 | 0.1915 |
| | Ours | **81.82** | 81.05 | **76.03** | 73.99 | **69.72** | **76.52** | **0.0308** | **0.0332** | **0.0287** | **0.0650** | **0.0504** | **0.0416** |
| MoE-LLaVA-Phi2-2.7B | APS | 79.51 | 82.14 | 72.74 | 74.51 | 66.56 | 75.89 | 0.2067 | 0.2863 | 0.2687 | 0.2476 | 0.3266 | 0.2672 |
| | LAC | 79.80 | 81.16 | 74.08 | 74.86 | 66.66 | 75.71 | 0.1377 | 0.2385 | 0.2212 | 0.2100 | 0.2825 | 0.2180 |
| | Ours | **81.62** | **83.06** | **74.44** | **75.86** | **69.02** | **76.80** | 0.0224 | 0.0991 | 0.0259 | 0.0333 | 0.0238 | 0.0409 |

Table 8: Evaluation of uncertainty quantification: Comparative analysis of the proposed CAP (Ours) method with standard Least Ambiguous Set-valued Classifiers (LAC) and Adaptive Prediction Sets (APS) methods (best values are in bold). The comparison includes different datasets and LLM models, with quality of uncertainty quantification evaluated using the Area Under the Receiver Operating Characteristic (AUROC) and the Area Under the Accuracy-Rejection Curve (AUARC).

| Model | Method | AUROC ↑ (Hallucination Detection) | | | | | | AUARC ↑ (Uncertainty guided selective generation) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HSwag | HDial | CQA | HSum | MMLU | Avg. | HSwag | HDial | CQA | HSum | MMLU | Avg. |
| Llama2-7B | APS | 0.4884 | 0.4646 | 0.6378 | 0.6353 | 0.4495 | 0.5351 | 0.3473 | 0.3301 | 0.5296 | 0.2962 | 0.5774 | 0.4161 |
| | LAC | 0.4079 | 0.2623 | 0.5490 | 0.7205 | 0.3594 | 0.4598 | 0.3395 | 0.2891 | 0.5185 | 0.2923 | 0.5496 | 0.3978 |
| | Ours | **0.7066** | **0.7040** | **0.7724** | **0.7672** | **0.6324** | **0.7165** | **0.8681** | **0.8354** | **0.9599** | **0.9078** | **0.8935** | **0.8929** |
| Llama2-13B | APS | 0.6225 | 0.3460 | 0.5186 | 0.4092 | 0.4132 | 0.4619 | 0.5788 | 0.5065 | 0.7893 | 0.4709 | 0.7455 | 0.6182 |
| | LAC | 0.4685 | 0.2007 | 0.6377 | 0.3478 | 0.3808 | 0.4071 | 0.5591 | 0.4801 | 0.7710 | 0.4580 | 0.6950 | 0.5926 |
| | Ours | **0.6396** | **0.5043** | **0.7159** | **0.6255** | **0.5572** | **0.6085** | **0.9254** | **0.8134** | **0.9685** | **0.8986** | **0.9177** | **0.9047** |

set size inversely correlates with model scale; larger models produce smaller sets, reflecting greater precision and reduced uncertainty. Additionally, AUROC and AUARC consistently improve with increasing model scale, indicating that larger models are more accurate, less prone to hallucinations, and better at abstaining when uncertainty is high.

Further, Figures 7 and 8 illustrate these gains for LLMs, showing larger models achieve higher accuracy and produce smaller set sizes while better avoiding hallucinations and performing uncertainty-guided selective generation. In Figure 6, we observe slight gains in all metrics when comparing VLMs with 7B parameters against Yi-VL with 6B parameters.

Table 9: Evaluation of coverage rate (%) and set size: Comparative analysis of the proposed CAP (Ours) method with standard LAC and APS methods. The comparison includes different datasets and LLM models, showcasing the satisfied coverage rate and balanced set sizes produced by our method with underlined values.

| Model | Method | Coverage (%) ↑ | | | | | | SS ↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HSwag | HDial | CQA | HSum | MMLU | Avg. | HSwag | HDial | CQA | HSum | MMLU | Avg. |
| Llama2-7B | APS | 90.02 | 90.44 | 91.78 | 89.72 | 92.50 | 90.89 | 3.346 | 3.257 | 2.661 | 3.227 | 3.319 | 3.162 |
| | LAC | 90.66 | 89.96 | 90.08 | 89.22 | 90.54 | 90.09 | 3.253 | 3.251 | 2.275 | 3.423 | 3.021 | 3.044 |
| | Ours | 90.38 | 90.42 | 91.22 | 89.78 | 91.04 | 90.56 | 3.378 | 3.252 | 2.316 | 3.360 | 3.191 | 3.099 |
| Llama2-13b | APS | 89.70 | 90.32 | 97.06 | 90.26 | 95.86 | 92.64 | 2.801 | 2.571 | 2.881 | 2.306 | 3.320 | 2.776 |
| | LAC | 89.88 | 90.62 | 90.52 | 89.98 | 89.18 | 90.03 | 2.497 | 2.535 | 1.568 | 2.117 | 2.578 | 2.259 |
| | Ours | 90.11 | 90.41 | 94.40 | 90.30 | 93.62 | 91.77 | 3.071 | 2.537 | 2.465 | 2.122 | 3.104 | 2.660 |

Table 10: Evaluation of accuracy (%) and ECE: Comparative analysis of the proposed CAP (Ours) method with standard LAC and APS methods. The comparison includes different datasets and LLM models, demonstrating a significant reduction in expected calibration error while improving overall accuracy.

| Model | Method | Accuracy (%) ↑ | | | | | | ECE ↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HSwag | HDial | CQA | HSum | MMLU | Avg. | HSwag | HDial | CQA | HSum | MMLU | Avg. |
| Llama2-7B | APS | 54.86 | 50.54 | 74.43 | 60.78 | 59.33 | 59.99 | 0.5720 | 0.5934 | 0.5085 | 0.6176 | 0.4894 | 0.5562 |
| | LAC | 55.05 | 50.24 | 74.10 | 59.23 | 59.11 | 59.55 | 0.5784 | 0.5927 | 0.4915 | 0.6127 | 0.4703 | 0.5491 |
| | Ours | **61.00** | **57.78** | **80.32** | **68.36** | **64.13** | **66.32** | **0.0606** | **0.0572** | **0.1953** | **0.1693** | **0.0414** | **0.1048** |
| Llama2-13b | APS | 70.38 | **66.84** | 83.42 | 72.59 | 65.48 | 71.74 | 0.4165 | 0.4207 | 0.3462 | 0.4666 | 0.3930 | 0.4086 |
| | LAC | 70.46 | 66.69 | 83.26 | 72.34 | 64.79 | 71.49 | 0.4183 | 0.4373 | 0.2808 | 0.4638 | 0.3444 | 0.3889 |
| | Ours | **73.31** | 64.28 | **85.89** | **73.44** | **68.11** | **73.00** | **0.0814** | **0.0721** | **0.1138** | **0.1559** | **0.0203** | **0.0887** |

However, since VLM size differences in this benchmark are minor, part of the performance gap may also stem from differences in finetuning methods and pre-trained models.

### D.5. Size Distribution of Predicted Set

The distribution of prediction types offers insight into our model's decision-making behavior across different vision-language tasks. LLaVA-1.6-34B favors set predictions across all benchmarks, with rates ranging from 55.4% (AI2D) to 62.4% (ScienceQA). This suggests the model often identifies multiple plausible answers rather than a single one due to underlying VLM response uncertainty. Single predictions comprise a substantial portion (31.5% to 38.8%), indicating high confidence in unique answers. Abstention rates vary notably across datasets, from 6.1% on ScienceQA to 12.8% on AI2D, reflecting the model's ability to recognize uncertainty in visual reasoning contexts. This pattern repeats for the Yi-34B LLM across five different tasks. This distribution demonstrates our selective prediction approach effectively captures different levels of model uncertainty, enabling more nuanced and reliable responses across diverse vision-language tasks.

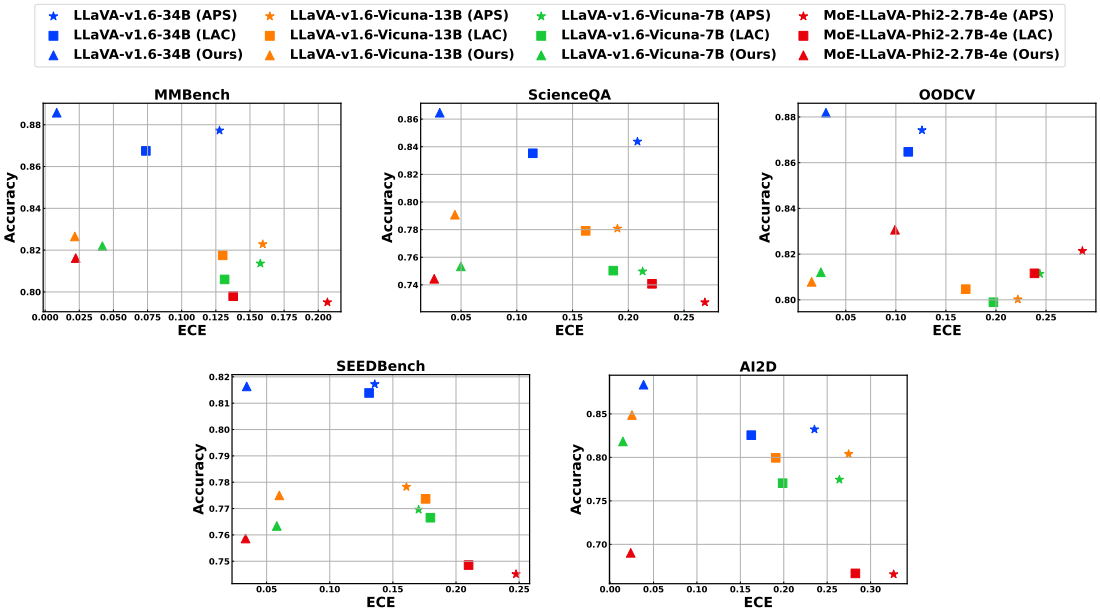


Figure 1: Accuracy vs. Expected Calibration Error (ECE) comparison of CAP, APS, and LAC across various VLMs and five datasets: MMBench, ScienceQA, OODCV, SEEDBench, and AI2D. An ideal model has high accuracy and low ECE (upper-left).

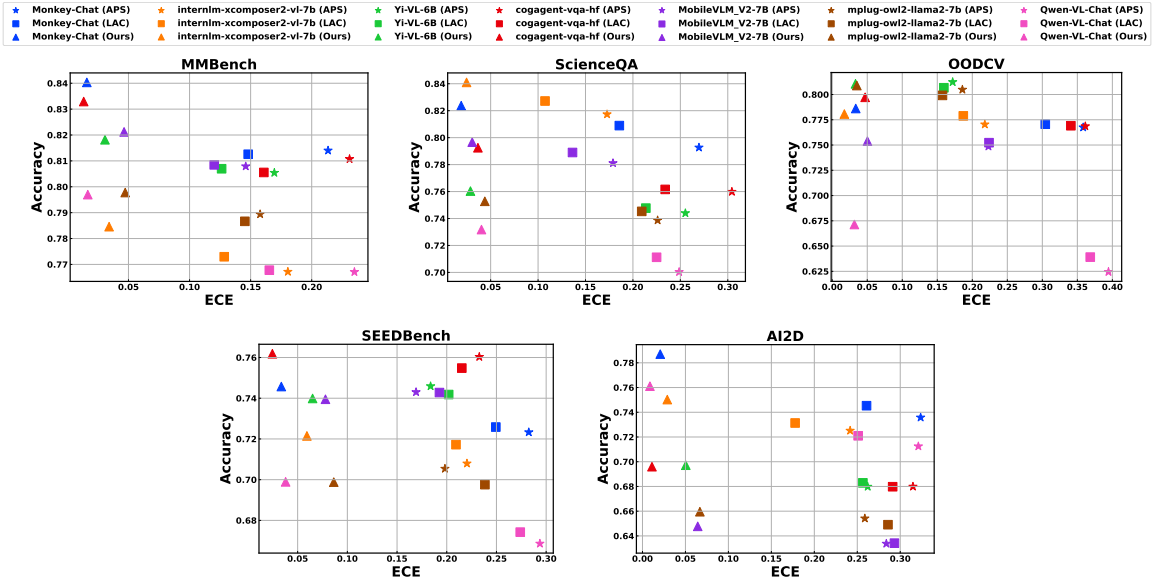


Figure 2: Accuracy versus Expected Calibration Error (ECE) comparison between CAP, APS and LAC methods across different VLMs and five datasets: MMBench, ScienceQA, OODCV, SEEDBench, AI2D. An ideal model has high accuracy and low ECE, indicating accurate predictions with well-calibrated uncertainty quantification (upper-left of the plot).

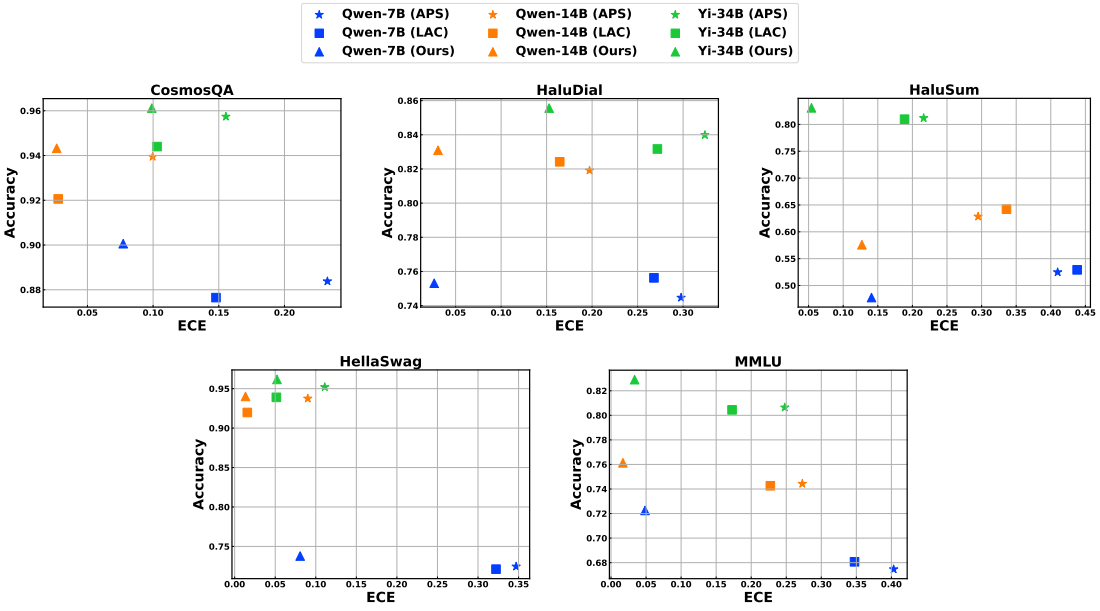


Figure 3: Accuracy versus Expected Calibration Error (ECE) comparison between CAP, APS and LAC methods across different LLMs and five datasets: CosmosQA, HaluDial, HaluSum, HellaSwag, MMLU. An ideal model has high accuracy and low ECE, indicating accurate predictions with well-calibrated uncertainty quantification (upper-left of the plot). The ECE of CAP shows significant improvement compared to baseline methods.

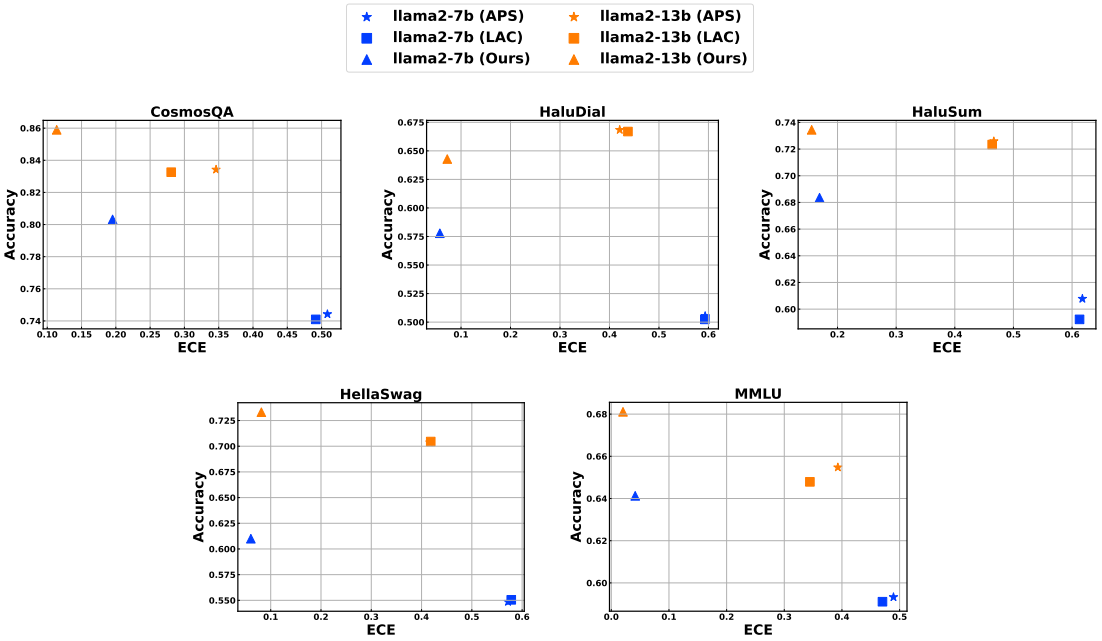


Figure 4: Accuracy versus Expected Calibration Error (ECE) comparison between CAP, APS, and LAC across five LLM datasets: CosmosQA, HaluDial, HaluSum, HellaSwag, and MMLU. Ideal models appear in the upper-left (high accuracy, low ECE). CAP consistently outperforms baselines in calibration quality.

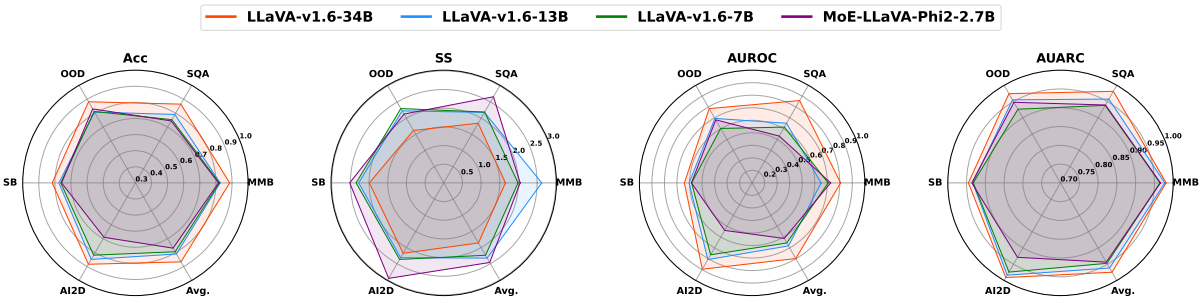


Figure 5: Performance comparison of VLMs with different model sizes (2.7B to 34B) across various metrics. Figures from left to right represent the performance of four models on one of four metrics: i) accuracy, ii) set size, iii) AUROC, and iv) AUARC. Each figure shows model performance across five VLM benchmark datasets and the effect of model scale (number of parameters) on different uncertainty metrics.

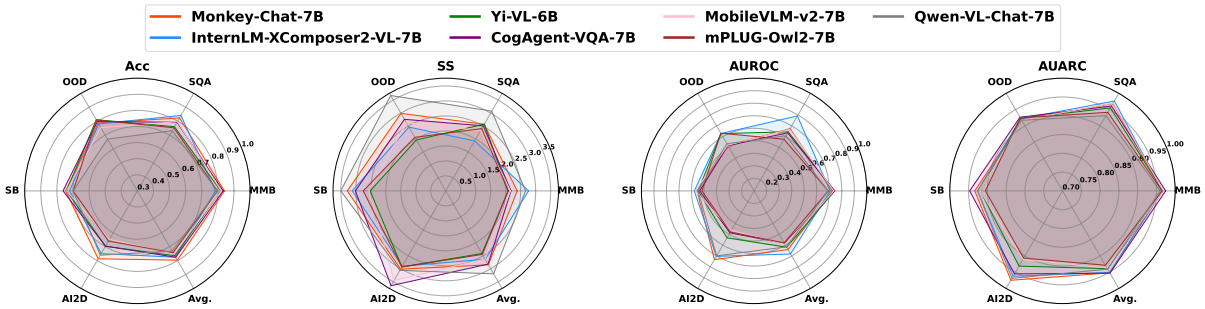


Figure 6: Performance comparison of additional VLMs with different model sizes (6B to 7B) across various metrics. Figures from left to right represent the performance of four models on one of four metrics: i) accuracy, ii) set size, iii) AUROC, and iv) AUARC. Each figure shows model performance across five VLM benchmark datasets and the effect of model scale (number of parameters) on different uncertainty metrics.

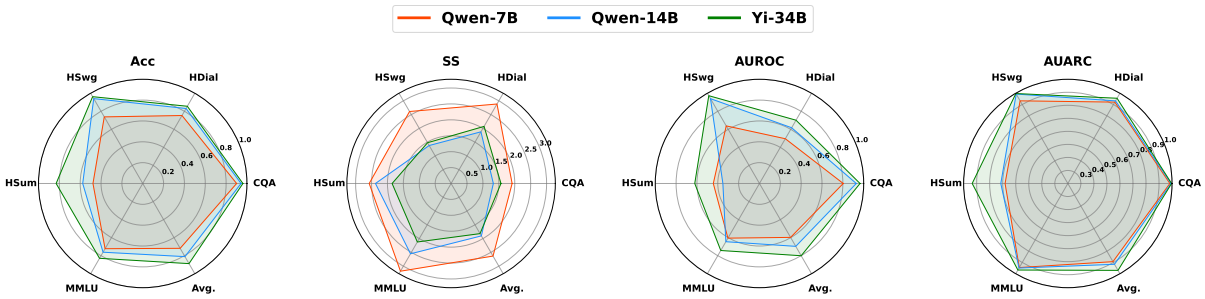


Figure 7: Performance comparison of LLMs with different model sizes (7B to 34B) across various metrics. Figures from left to right represent the performance of four models on one of four metrics: i) accuracy, ii) set size, iii) AUROC, and iv) AUARC. Each figure shows model performance across five LLM benchmark datasets and the effect of model scale (number of parameters) on different uncertainty metrics.

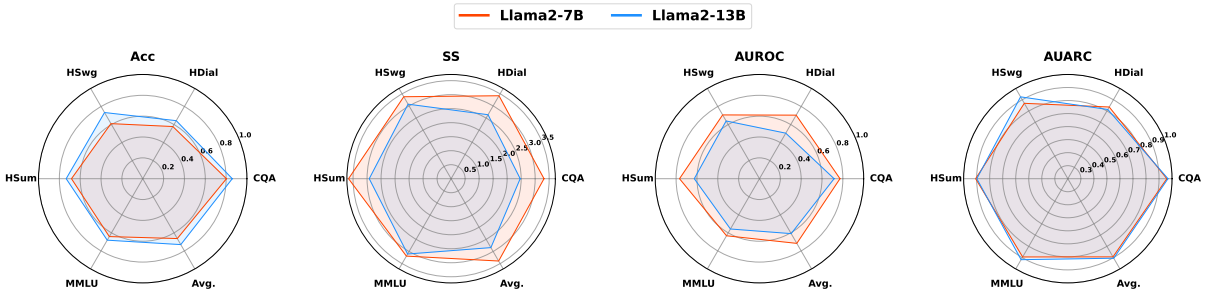


Figure 8: Performance comparison of Llama-2 series LLMs with different model sizes (7B and 13B) across various metrics. Figures from left to right represent the performance of two models on four metrics: i) accuracy, ii) set size, iii) AUROC, and iv) AUARC. Each figure shows model performance across five LLM benchmark datasets and the effect of model scale (number of parameters) on different uncertainty metrics.