**What is Chameleon?**

Chameleon is a new kind of model created by the team at FAIR, Meta. It can understand and create both images and text together in any order. Here’s a simplified summary:

**Introduction**

- **What It Does**: Chameleon can handle tasks like answering questions about images, describing pictures, writing text, and creating images, all with one model.

- **Performance**: It performs very well, even better than some specialized models in tasks like image captioning and text generation.

- **Unified Approach**: Unlike many models that handle text and images separately, Chameleon processes them together from the start. This helps it understand and generate complex documents with mixed content.

- **Training and Innovations**: The team developed special techniques to train Chameleon effectively, ensuring it can handle mixed content smoothly.

- **Human Evaluation**: In tests where people compared its outputs, Chameleon was often preferred over other leading models like Gemini-Pro and GPT-4V.

**Key Features of Chameleon**

1. **Early Fusion**: Combines text and image processing right from the beginning, allowing seamless integration.

2. **Token-Based Representation**: Uses tokens for both text and images, treating them similarly within the model.

3. **Transformer Architecture**: Applies the same model structure to both image and text tokens.

4. **Stable Training**: Special modifications ensure the model trains well without issues.

5. **High Performance**: Excels in a variety of tasks, from visual question answering to text generation.

**Tasks and Evaluation**

- **Visual Question Answering**: Performs better than other models like Flamingo and Llava-1.5.

- **Image Captioning**: Generates descriptions of images very accurately.

- **Text Generation**: Competes well with text-only models like Llama-2.

- **Image Generation**: Can create images from text prompts, adding visual elements to documents.

- **Mixed-Modal Tasks**: Excels in tasks that require handling both images and text together.

**Pre-Training**

**Overview:**

Chameleon works with both text and images by breaking them into discrete tokens. It uses the scalability of auto-regressive Transformers, a type of AI model. This process is based on previous research.

**Training Method:**

We train the model using different combinations of text and images:

- Text-only

- Single text/image pairs

- Fully mixed text-image documents

Example Task:

Prompt: "



While chameleons are masters of camouflage in nature, let's see how they might fare in a man-made setting."

1. Analyze the colors, patterns, and textures in this image and identify the most challenging spot for a chameleon to blend in.

2. Generate an image of a chameleon with the best pattern to camouflage itself.

3. Rate and explain how successful each attempt is on a scale from 1-10.

**Response Example:**



The image shows a stop sign in a forest with sunlight filtering through trees. The stop sign, which is bright red with white letters, stands out against the green leaves and trees.

- **Challenging Spot:** The stop sign is the hardest spot for a chameleon to blend in due to its bright red color and bold white letters.

- **Optimal Pattern:** To blend in, the chameleon would need shades of brown and red with a mottled pattern to scatter light and reduce visibility.

- **Rating:** 2/10. The chameleon’s green and brown pattern helps, but the stop sign's bright red color still makes it stand out.

**Image Tokenization:**

We train a tokenizer that converts a 512x512 image into 1024 tokens. This tokenizer struggles with images that have a lot of text. To improve, we double the amount of images with human faces during training.

**Text Tokenization:**

We also train a Byte Pair Encoding (BPE) tokenizer with a vocabulary of 65,536 tokens, including the 8192 image tokens.

**Example Recipe Prompt:**

Prompt: "What can I bake with this? Give me the full recipe and show me an image of the final dish."

Response:

"Bananas are versatile for baked goods like muffins, bread, cakes, and cookies. Here’s a simple banana bread recipe using 3 ripe bananas."

**Ingredients:**

- 3 ripe bananas, mashed

- 1 1/2 cups all-purpose flour

- 1 tsp baking powder

- 1/2 tsp baking soda

- 1/2 tsp salt

- 1/4 cup unsalted butter, softened

- 1 cup granulated sugar

- 2 large eggs

- 1 tsp vanilla extract

- 1/2 cup chopped walnuts (optional)

**Instructions:**

1. Preheat the oven to 350°F (180°C). Grease a 9x5-inch loaf pan.

2. In a bowl, whisk together flour, baking powder, baking soda, and salt.

3. In a large bowl, cream butter and sugar until light and fluffy.

4. Beat in the eggs one at a time, then add mashed bananas and vanilla extract.

5. Gradually mix in dry ingredients until just combined.

6. Fold in walnuts if using.

7. Pour batter into the pan and smooth the top.

8. Bake for 50-60 minutes until a toothpick comes out clean.

9. Cool in the pan for 10 minutes, then transfer to a wire rack to cool completely.

**Training Data:**

The training process is divided into two stages.

- **First Stage:** Uses large unsupervised datasets including 2.9 trillion text tokens and 1.4 billion text-image pairs.

- **Second Stage:** Incorporates higher quality datasets and instruction tuning sets.

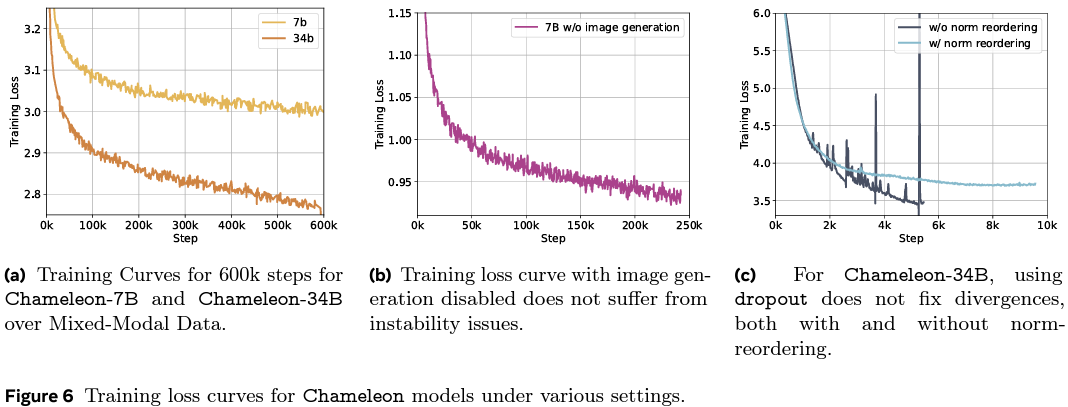
**Challenges and Solutions:**

Maintaining stability while scaling the model was tough. The Chameleon models above 8 billion parameters and 1 trillion tokens often became unstable late in training. To address this:

- We used Query-Key normalization (QK-Norm) to manage norm growth in the model.

- For the 7 billion parameter model (Chameleon-7B), we added dropout after attention and feed-forward layers.

- For the 34 billion parameter model (Chameleon-34B), we used a different normalization order.



**Optimization:**

The models were trained using the AdamW optimizer with specific settings to ensure stability. Dropout and z-loss regularization were also used for the smaller model, while only z-loss was used for the larger model.

**Inference:**

Our inference pipeline is designed to handle the unique challenges of generating both text and images. It supports streaming to improve efficiency and reduce latency.

**Summary of Training Resources:**

The training utilized Meta's Research SuperCluster with NVIDIA A100 80GB GPUs. Here’s the resource usage:

- **Chameleon-7B:** 1024 GPUs for 856,481 GPU hours

- **Chameleon-34B:** 3072 GPUs for 4,282,407 GPU hours

**Alignment**

**Alignment Process**

We follow recent methods for aligning models using supervised fine-tuning on high-quality datasets (Zhou et al., 2023). We use various types of data to showcase the model's abilities and enhance its safety.

**Data Categories**

Our supervised fine-tuning (SFT) data is divided into:

- Text

- Code

- Visual Chat

- Image Generation

- Interleaved Text/Image Generation

- Safety

Examples of each category from the Chameleon-SFT dataset are shown in Figure 7. We get our Text SFT data from LLaMa-2 (Touvron et al., 2023) and Code SFT data from CodeLLaMa (Roziere et al., 2023).

For Image Generation, we use highly aesthetic images filtered by an aesthetic classifier (Schuhmann et al., 2022). We select images rated at least six and pick the top 64,000 closest to the size 512 × 512.

For Visual Chat and Interleaved Text/Image Generation, we collect high-quality data using third-party vendors, as recommended by Touvron et al. (2023) and Zhou et al. (2023). We do not use any Meta user data. The dataset statistics are shown in Table 3.

**Safety Data**

We include prompts that could make the model generate unsafe content, and pair them with refusal responses (e.g., "I can't help with that"). These prompts cover sensitive topics like violence, drugs, privacy, and sexual content. Our safety data includes examples from LLaMa-2-Chat (Touvron et al., 2023), synthetic examples from Rainbow Teaming (Samvelyan et al., 2024), image generation prompts from Pick-A-Pic (Kirstain et al., 2023), cybersecurity safety examples (Roziere et al., 2023), and mixed-modal prompts collected internally (Honovich et al., 2022). Mixed-modal prompts are important for addressing potential multi-modal attacks.

**Fine-Tuning Strategy**

**Data Balancing**

Balancing different types of data during SFT is crucial for quality alignment. An imbalance can cause the model to favor one type of data too much.

**Optimization**

Our fine-tuning strategy uses a cosine learning rate schedule, starting at 1e-5, with a weight decay of 0.1. We use a batch size of 128 and sequences up to 4096 tokens. Each dataset instance includes a prompt and its answer. We pack as many prompts and answers as possible into each sequence, using a special token to separate prompts and answers. We focus on optimizing the model based on the answer tokens, which improves performance slightly. We also use a dropout rate of 0.05.

During fine-tuning, prompt images are resized with border padding to keep all information, while answer images are center-cropped for better visual quality.

**Human Evaluations and Safety Testing**

Chameleon has advanced capabilities in understanding and generating mixed-modal content (text and images), which are difficult to measure with current benchmarks. Here’s how we evaluate its performance through human assessments and safety tests.

**Collecting Prompts for Evaluation**

We collaborated with a third-party crowdsourcing service to gather a diverse set of prompts from human annotators. Annotators were asked to think creatively about what they would want a multi-modal model to generate in various real-life scenarios. For example, in a kitchen scenario, they might ask, "How to cook pasta?" or "Show me examples of kitchen island designs."

These prompts included both text-only and text-with-images formats. We then had three random annotators review the prompts for clarity and whether they required mixed-modal responses. Based on majority votes, unclear prompts and those not needing mixed-modal responses were filtered out, leaving us with 1,048 prompts: 441 (42.1%) mixed-modal and 607 (57.9%) text-only.

**Baselines and Evaluations**

We compared Chameleon 34B with OpenAI GPT-4V and Google Gemini Pro by using their APIs. While these models can accept mixed-modal prompts, their responses are text-only. To strengthen the baselines, we enhanced GPT-4V and Gemini by having them generate image captions, which were then used to create images with OpenAI DALL-E 3. These enhanced responses are referred to as GPT-4V+ and Gemini+.

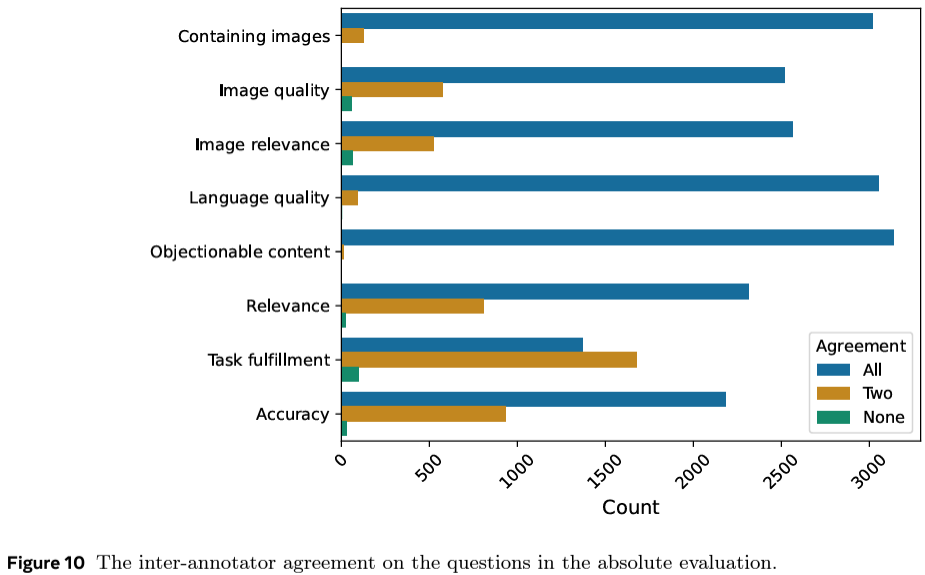
Using the same crowdsourcing service, we conducted two types of evaluations:

1. **Absolute Evaluation**: Each model’s response was judged separately by three annotators based on relevance and quality. Annotators assessed whether the response fully, partially, or did not fulfill the task described in the prompt. Chameleon performed better, with 55.2% of its responses fully fulfilling the tasks, compared to 37.6% for Gemini+ and 44.7% for GPT-4V+.

2. **Relative Evaluation**: Annotators compared Chameleon’s responses directly with those of the baseline models. Chameleon was preferred over Gemini+ in 41.5% of cases and over GPT-4V+ in 35.8% of cases.

**Inter-Annotator Agreement**

To ensure the quality of evaluations, we analyzed the agreement among annotators. For simple, objective questions, there was high agreement, particularly regarding the presence of objectionable content. For more subjective questions, such as task fulfillment, there was less agreement, but disagreements were usually close rather than stark.



**Safety Testing**

We tested Chameleon's ability to handle unsafe content by using crowdsourced prompts designed to provoke harmful responses (e.g., self-harm, violence). Annotators labeled responses as safe, unsafe, or unsure. The majority of Chameleon's responses were safe, with very few unsafe responses: 0.39% for the 7B model and 0.095% for the 30B model. Additionally, internal testing with adversarial prompts showed that only 1.6% of the responses were unsafe.

**Discussion**

Chameleon excels in generating mixed-modal responses that are relevant and appealing. However, there are limitations in our evaluation, such as the use of crowdsourced rather than real-user prompts, and the exclusion of certain visual tasks like OCR. Future comparisons would benefit from using other native mixed-modal models instead of enhancing text-only responses from existing models.

Overall, Chameleon shows strong performance and safety in generating mixed-modal content, making it a competitive choice in the realm of multi-modal AI systems.

**Benchmark Evaluations**

**Text Evaluation**

Since Chameleon has broad capabilities, we can't compare it directly to just one model. Instead, we compare it to the best models in various categories.

**Text-Only Models**

We evaluated our pre-trained text-only model (not fine-tuned) against other leading text-only language models using an in-house evaluation platform. The evaluation covered areas like commonsense reasoning, reading comprehension, math problems, and world knowledge. The results are summarized in Table 6.

**Performance Comparison:**

- **Commonsense Reasoning and Reading Comprehension**: We measured 0-shot performance on several benchmarks:

- **PIQA**: Chameleon-34B outperformed Llama-2 70B.

- **SIQA and HellaSwag**: Chameleon models were competitive.

- **WinoGrande**: Chameleon-34B surpassed Llama-2 70B.

- **ARC-Easy, ARC-Challenge, OpenBookQA, BoolQ**: Chameleon-34B performed well, often beating or matching larger models.

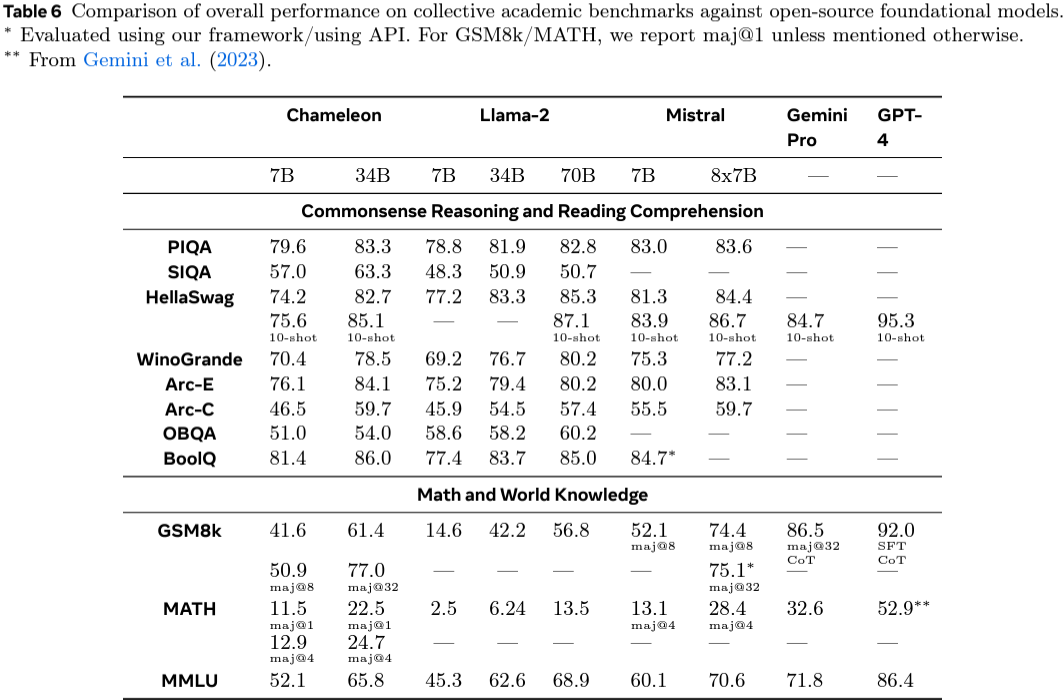
**Math and World Knowledge:**

- **GSM8k** (grade school math problems): Chameleon-7B outperformed Llama-2 models and was on par with Mistral 7B.

- **MATH**: Chameleon-34B outperformed Llama-2 70B.

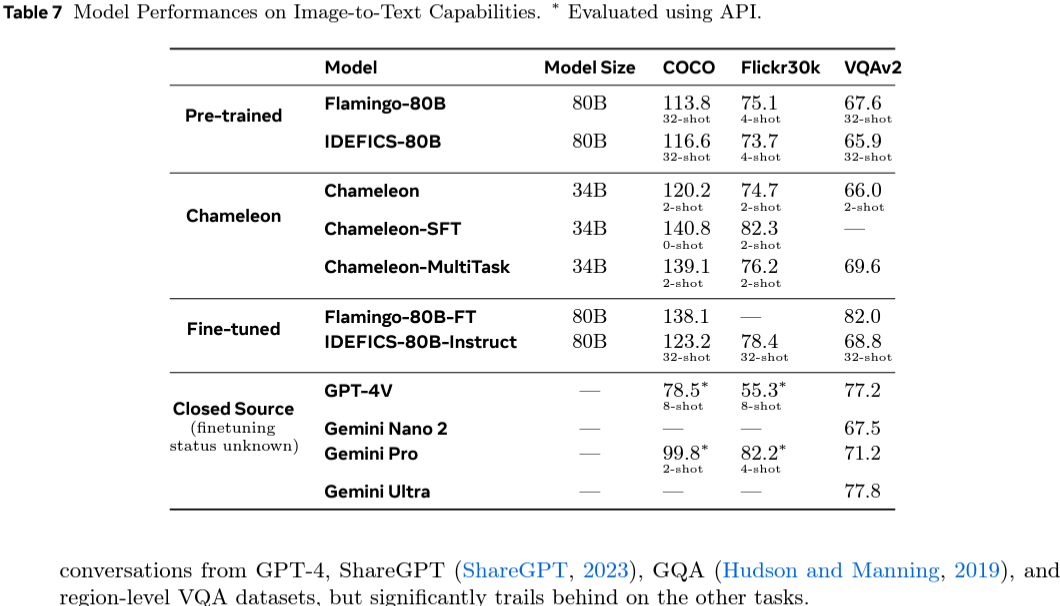
- **MMLU** (world knowledge): Both Chameleon models outperformed Llama-2, with Chameleon-34B approaching the performance of advanced models like Mixtral 8x7B.

Chameleon's success is likely due to factors like using more pre-training compute, including code data, and having high-quality data in the final stages of training.



**Image-to-Text Evaluation**

We tested Chameleon on tasks requiring text generation from images, such as image captioning and visual question answering (VQA).



**Image Captioning**:

- **MS-COCO and Flickr30k datasets**: Chameleon-34B outperformed larger models like Flamingo 80B and IDEFICS 80B using fewer in-context examples.

**Visual Question Answering**:

- **VQA-v2 dataset**: Chameleon-34B's pre-trained model matched the performance of larger models using fewer examples. The fine-tuned Chameleon models were competitive with leading models like Flamingo-80B-FT and GPT-4V.

Overall, Chameleon performs well across both text-only and image-to-text tasks, often rivaling or surpassing larger models with fewer training examples.