

---

# Escaping the SpuriVerse: Can Large Vision-Language Models Generalize Beyond Seen Spurious Correlations?

---

1 This document supplements the main paper with additional details. Below is the outline:

- 2 • Section A details the prompts and annotation interface used for curating SpuriVerse.
- 3 • Section B details the prompts used, fine-tuning and evaluation details, and compute resources
- 4 for experiments.
- 5 • Section C discusses the limitations of the work.
- 6 • Section D discusses the societal impacts of the work.
- 7 • Section E reports the licenses and terms of use for the models and datasets.

## 8 A SpuriVerse Details

### 9 A.1 Prompts used for Dataset Curation

10 In this section, we include the prompts used for each step in the pipeline.

11 **Step 1: Select a challenging set.** The following template was used to prompt GPT-4o to find the  
12 error set on each source benchmark.

(System Prompt)

You will be given an image, and a multiple choice question regarding the image. You will provide your answer as one of the options (A), (B), (C), or (D). You will answer correctly. You will not use any fullstops or punctuation. You will not explain your answer or write words before or after the answer. Only the answer itself will you respond with.

(User Prompt)

```
<Image/>
Question: {sample['question']}
Options:
(A) {sample['A']}
(B) {sample['B']}
(C) {sample['C']}
(D) {sample['D']}
Please select the correct answer from the options above.
```

13 **Step 2: Two-stage verification.** The following prompt was used to determine whether a sample falls  
14 into *VLM Accepted* subset.

(System Prompt)

You are a helpful assistant to determine if a model’s error is caused primarily by spurious correlations, patterns that can often be used to predict the target, but are not actually causal.

(User Prompt)

<Image/>

Given this image, a Large Multi-modal Model was asked, sample[‘question’], and given the choices:

(A) {sample[‘A’]}

(B) {sample[‘B’]}

(C) {sample[‘C’]}

(D) {sample[‘D’]}

The model chose {prediction} and the correct answer is {answer}. The error is most likely due to spurious correlation. List the top two spurious attributes that the model may have used to predict the wrong answer {prediction}.

- 15 **Step 3: Generate counterfactual scene descriptions.** The following prompt was used to generate the pairs of counterfactual scene descriptions. Specifically, the goal is to first construct a description that enables the question-answer pair to hold in the scene description. Then, generate a spurious counterpart by extending the previous description to contain the spurious attributes provided.

<Image/>

You are given the following:

question: {question}

answer: {answer}

spurious attribute: {attributes}

Based on the question and the answer, generate a description of a scene such that when the question is asked, the answer is {answer}. Keep the description to one short sentence.

Write another one sentence description that includes the spurious attribute while maintaining the same context.

Return the response in JSON format with the two keys: "positive" and "negative" where "positive" describes the scene with the spurious attribute and "negative" describes the scene without the spurious attribute.

- 20 **Step 5: Verify by Core vs. Spurious.** The same prompt in Step 1 was used to evaluate GPT-4o, Gemini 2.0 Flash, and Qwen-VL-Max on *Human Accepted* subset.

## 22 A.2 Annotation Interface for Dataset Curation

- 23 Figure 1 displays the interface for refining spurious attributes and image descriptions. In step 1, the interface displays the image and question from the error set. In addition, it also displays GPT-4o’s prediction and the ground truth answer. In step 2, annotators can view the prefetched spurious attributes by clicking on *From store* or generate new spurious attributes by clicking on *Run*. The annotators can then refine the spurious attributes in the text box. In step 3, annotators can click on *Run* and generate image descriptions for both spurious and core groups based on the spurious attributes extracted in Step 2. In step 4, the images for spurious and core groups can be generated

30 based on the descriptions in step 3. Annotators can go back to step 3 and refine the descriptions if the  
 31 images generated are not faithful. In step 5, annotators can take a peek at models' evaluations on one  
 32 image for spurious group and one image for core group.

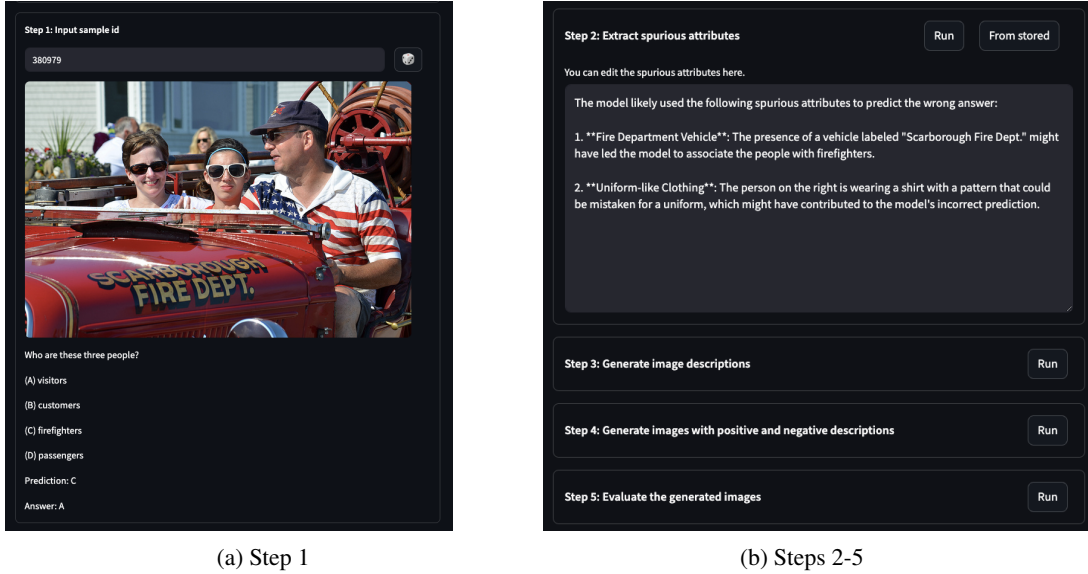


Figure 1: **Annotation interface** for refining spurious attributes and image descriptions. The curation pipeline consists of 5 steps: Step 1 displays the image and question from the error set. Step 2 allows annotators to view extracted spurious attributes by clicking on *From store* or generating new spurious attributes by clicking on *Run*. Annotators can edit the spurious attributes. Step 3 generates image descriptions based on the extracted or edited spurious attributes. Step 4 then generates images for spurious and core groups. Annotators can go back to step 3 and refine the description if the images generated are not faithful. Step 5 allows annotators to take a peek at models' evaluations on a few images (One for each group).

### 33 A.3 Prompt used for Category Identification

```
<spurious examples.csv/>
```

You are given a list of examples that a model answers incorrectly due to spurious correlation. For each row, a model makes an error due to the correlation between the spurious attribute (spuriousAttr) and the prediction. If the prediction is 'Yes' or 'No', refer to the question for context.

Determine the spurious correlation for each row. Then create a taxonomy based on the spurious correlations.

### 34 A.4 Dataset Cost

35 We used Stable Diffusion to generate the synthetic images for verifying the spurious correlations,  
 36 and the spurious groups also make up part of the dataset. In our curation pipeline, 194 samples were  
 37 being selected after the human-VLM verification step. We thus generated  $194 \times 2 \times 10 = 3880$   
 38 images using Stable Diffusion Ultra. Since generating each image costs 8 credits, and each credit  
 39 costs \$0.01, the total cost is  $3880 \times 8 \times 0.01 = 310.4$  dollars.

## 40 B Experiments

### 41 B.1 Prompts used for Main results

42 We used the following prompt:

(System Prompt)

You will be given an image, and a multiple choice question regarding the image. You will provide your answer as one of the options (A), (B), (C), or (D). You will answer correctly. You will not use any fullstops or punctuation. You will not explain your answer or write words before or after the answer. Only the answer itself will you respond with.

(User Prompt)

<Image/>

Question: {sample['question']}

Options:

(A) {sample['A']}

(B) {sample['B']}

(C) {sample['C']}

(D) {sample['D']}

Please select the correct answer from the options above.

### 43 B.2 Prompts used for Prompting strategies

44 We kept the user prompt the same as Main Results. For Chain-of-thought, we used the following  
45 system prompt:

(System Prompt)

You will be given an image, and a multiple choice question regarding the image. Think step by step and give a final answer. You will include one of the choices (A), (B), (C), or (D) in your final answer.

46 For Spurious Aware, we used the following sytem prompt:

(System Prompt)

You will be given an image, and a multiple choice question regarding the image. Be aware that there may be some spurious features in the image that associate with some of the options. Describe the potential spurious features. Then give a answer without using the spurious features. You will include one of the choices (A), (B), (C), or (D) in your final answer.

### 47 B.3 Models

48 We evaluate SpuriVerse on 15 recent LVLMs, including GPT-4o [OpenAI, 2023], GPT-4o-mini [OpenAI, 2023], o4-mini [OpenAI, 2023], o3 [OpenAI, 2023], Gemini 2.0 Flash [DeepMind, 2024],  
49 Gemini 1.5 Pro [DeepMind, 2024], Claude 3.7 Sonnet [Anthropic, 2024], Qwen-VL-Max [Cloud, 2025],  
50 Qwen-VL-Pro [Cloud, 2025], Qwen2.5-vl-7b-instruct [Team, 2025], Qwen2.5-vl-32b-instruct [Team, 2025],  
51 Llama-3.2-11B-vision-instruct [Grattafiori et al., 2024], Llama-3.2-90B-  
52

vision-instruct [Grattafiori et al., 2024], LLaVA-v1.6 7b [Liu et al., 2023a] and LLaVA-v1.5 [Liu et al., 2023b].

We used OpenAI API [OpenAI, 2023] for making requests to GPT-4o, GPT-4o-mini, o4-mini, o3. We used Gemini API [DeepMind, 2024] for making requests to Gemini 2.0 Flash, Gemini 1.5 Pro. We used Anthropic API [Anthropic, 2024] for making requests to Claude 3.7 Sonnet. We used Qwen API [Cloud, 2025] for making requests to Qwen-VL-Max, Qwen-VL-Pro. We accessed Qwen2.5-vl-7b-instruct, Qwen2.5-vl-32b-instruct, Llama-3.2-11B-vision-instruct, Llama-3.2-90B-vision-instruct via Unsloth AI [Daniel Han and team, 2023]. We accessed LLaVA-v1.6 7b and LLaVA-v1.5 via Hugging Face [Wolf et al., 2020].

**Hyperparameters** During evaluation, for the reasoning models (o3 and o4-mini), we set max\_tokens to 3000. For all other models, we set max\_tokens to 300. All the open-sourced models are 4-bit quantized during evaluation.

**Versions** For GPT-4o, we used version “gpt-4o-2024-08-06”. For GPT-4o-mini, we used version “gpt-4o-mini-2024-07-18”. For Claude 3.7 Sonnet, we used version “claude-3-7-sonnet-20250219”.

## B.4 Finetuning details

We divided both the anchor set and the spurious groups into train/val/test sets according to the ratio of 70/10/20. We finetuned Llama-3.2-11B-vision-instruct and Qwen2.5-vl-7b-instruct on the train and val sets of anchors and spurious groups, respectively, and evaluated on the test sets. As a baseline, we also considered the “non-spurious set”, which was sampled randomly from the source benchmarks. The samples are drawn with the same benchmark distribution as the anchor set. Similarly, the non-spurious set is further split according to the ratio of 70/10/20. We also finetuned on the “Mixed set”, which was the concatenation of spurious groups and the “non-spurious set”.

All finetuning results were measured across 5 seeds, where each seed corresponds to a different split of the data.

We finetuned both Llama-3.2-11B-vision-instruct and Qwen2.5-vl-7b-instruct using unsloth [Daniel Han and team, 2023].

We used the following hyperparameters for both models:

```
finetune_vision_layers=True,
finetune_language_layers=True,
finetune_attention_modules=True,
finetune_mlp_modules=True,
r=16,
lora_alpha=16,
lora_dropout=0,
bias="none",
random_state=3407,
use_rslora=False,
loftq_config=None
```

We used the following SFT configuration for both models:

```
per_device_train_batch_size=2,
```

```

gradient_accumulation_steps=4,
warmup_steps=5,
num_train_epochs=10,
learning_rate=2e-4,
logging_steps=1,
optim="adamw_8bit",
weight_decay=0.01,
lr_scheduler_type="linear",
seed=3407,
remove_unused_columns=False,
dataset_text_field="",
dataset_kwargs={"skip_prepare_dataset": True},
dataset_num_proc=4,
max_seq_length=2048,
eval_strategy="epoch",
load_best_model_at_end=True,
save_strategy="best",
metric_for_best_model="eval_loss",
greater_is_better=False,
save_total_limit=2,

```

We used the following instruction format during finetuning:

```

<Image/>
Question:  sample['question']
Options:
(A) sample['A']
(B) sample['B']
(C) sample['C']
(D) sample['D']
Please select the correct answer from the options above.

```

## B.5 Robustness-accuracy Tradeoff

The procedure to replace spurious samples with non-spurious samples is described in Algorithm 1. Particularly, we use the anchor set  $\mathcal{S}_{\text{anchor}}$  as the reference to decide the distribution of the source benchmarks in the training set. If a sample  $s_i$  in  $\mathcal{S}_{\text{anchor}}$  is determined to use non-spurious samples, then we randomly sample 10 images from the source benchmark  $s_i$  originates from. Otherwise, we use the original spurious group images  $G_i$  of size 10.

As a toy example, suppose  $\mathcal{S}_{\text{anchor}} = \{s_1, s_2, s_3, s_4, s_5\}$ . Let  $\{s_1, s_2, s_3, s_4\}$  be the training split at first, and each of them comes from a distinct benchmark we use. Let the second split between spurious and non-spurious yields  $\mathcal{S}_{\text{train, spurious}} = \{s_1, s_2\}$ ,  $\mathcal{S}_{\text{train, non-spurious}} = \{s_3, s_4\}$  with  $r = 50\%$ . Then, the next step will yield a training set of 40 samples, consisting of 20 samples from  $G_1, G_2$ , 10 samples drawn from the source benchmark of  $s_3$ , and 10 samples drawn from the source benchmark of  $s_4$ . With this training set  $\mathcal{S}'_{\text{train}}$ , we then do a train-val split with a fraction of 87.5% and 12.5%.

## B.6 Compute Resources

The experiments were conducted on a 4xNVIDIA H100, where the GPU memory is 4x95830MB. The CPU architecture is x86\_64, and there are 64 CPUs.

For the finetuning experiments, since both the Llama-3.2-11B-vision-instruct and Qwen2.5-vl-7b-instruct are 4-bit quantized and optimized with UnslothAI, approximately 12GB of GPU memory is sufficient for finetuning. Finetuning each model on spurious groups/non-spurious groups for 10 epochs takes about 3 hours on a single H100. Overall, the total compute time is  $\text{num\_setups}(3) \times \text{num\_models}(2) \times \text{num\_seeds}(5) \times \text{duration}(3) = 90\text{hours}$ . For the accuracy-robustness trade-

---

**Algorithm 1** Sampling procedure to replace spurious samples with non-spurious samples

---

**Input:** anchor set  $\mathcal{S}_{\text{anchor}}$   
**Input:** spurious fraction  $r$   
**Input:** spurious group samples  $G$   $\triangleright G_i$  is a set of 10 samples using synthetic images for  $s_i \in \mathcal{S}_{\text{anchor}}$   
**Input:** benchmark samples  $B$   $\triangleright B_i$  is the set of all samples from the  $i$ -th benchmark

Let  $b(s)$  be the source benchmark of a sample  $s$   
 $\mathcal{S}_{\text{train}}, \mathcal{S}_{\text{test}} \leftarrow \text{random\_split}(\mathcal{S}_{\text{anchor}}, \text{fraction}=[0.8, 0.2])$   
 $\mathcal{S}_{\text{train, spurious}}, \mathcal{S}_{\text{train, non-spurious}} \leftarrow \text{random\_split}(\mathcal{S}_{\text{train}}, \text{fraction}=[r, 1-r])$   
 $\mathcal{S}'_{\text{train}} \leftarrow \{\}$   
**for**  $s_i \in \mathcal{S}_{\text{train, spurious}}$  **do**  
     add  $G_i$  to  $\mathcal{S}'_{\text{train}}$   
**end for**  
**for**  $s_i \in \mathcal{S}_{\text{train, non-spurious}}$  **do**  
      $j \leftarrow b(s_i)$   
      $S_{s_i, B_j} \sim B_j$  where  $|S_{s_i, B_j}| = 10$   $\triangleright$  Draw 10 random samples from the source benchmark  
     add  $S_{s_i, B_j}$  to  $\mathcal{S}'_{\text{train}}$   
**end for**  
 $\mathcal{S}''_{\text{train}}, \mathcal{S}''_{\text{val}} \leftarrow \text{random\_split}(\mathcal{S}'_{\text{train}}, \text{fraction}=[0.875, 0.125])$   $\triangleright$  fraction to make train/val/test split 70/10/20 w.r.t.  $\mathcal{S}_{\text{anchor}}$   
**Output:**  $\mathcal{S}''_{\text{train}}, \mathcal{S}''_{\text{val}}, \mathcal{S}_{\text{test}}$

---

104 off experiment, the total compute time is  $\text{num\_setups}(6) \times \text{num\_models}(2) \times \text{num\_seeds}(5) \times$   
 105  $\text{duration}(3) = 180\text{hours}$ . In this case, the setup refers to the proportion of the spurious samples.

106 Finetuning on the anchor set takes approximately 0.5 hour, since the anchor set is a much smaller  
 107 set. Hence the total compute time is  $\text{num\_setups}(3) \times \text{num\_models}(2) \times \text{num\_seeds}(5) \times$   
 108  $\text{duration}(0.5) = 15\text{hours}$ .

109 For the main results, running Llama-3.2-90B-Vision-Instruct for inference takes about 45 GB GPU  
 110 memory, and running Qwen2.5-vl-32b-instruct takes about 24 GB GPU memory. All the other  
 111 open-source models can be run under 12 GB of GPU memory. Evaluation of each non-reasoning  
 112 model takes about 0.5 hour on spurious and non-spurious samples. Hence, the total compute time  
 113 is  $\text{num\_setups}(2) \times \text{num\_models}(13) \times \text{duration}(0.5) = 13\text{hours}$ . Evaluation of the reasoning  
 114 models (o3 and o4-mini) takes about 10 hours. Hence, the total compute time is  $\text{num\_setups}(2) \times$   
 115  $\text{num\_models}(2) \times \text{duration}(10) = 40\text{hours}$ .

116 Evaluating on the anchor set takes about 0.05 hours for the non-reasoning models, and 1 hour for the  
 117 reasoning models. Hence, the total compute time for non-reasoning models is  $\text{num\_setups}(2) \times$   
 118  $\text{num\_models}(13) \times \text{duration}(0.05) = 1.3\text{hours}$ , and the total compute time for reasoning models  
 119 is  $\text{num\_setups}(2) \times \text{num\_models}(2) \times \text{duration}(1) = 4\text{hours}$ .

120 The full research project does not require more compute than the experiments reported in the paper.

## 121 C Limitations

122 **Curation Bias.** Two forms of bias can exist in our curation pipeline. First, we use GPT-4o as the  
 123 single strong model in the early steps of the curation pipeline. This procedure can limit us to spurious  
 124 correlations closer to its training distribution. However, our last counterfactual verification attempts  
 125 to mitigate this potential bias. Secondly, the human annotations were done by two contributors on the  
 126 team with agreement. There can be annotation variance when it comes to other annotators.

127 **Sample Format.** In the collection of existing benchmarks, we only use multiple-choice question-  
 128 answering benchmarks because they limit the output space compared to open-generation format,  
 129 allowing easier investigation into why a model makes an incorrect prediction over the correct answer.  
 130 Indeed, spurious correlation can exist when the spurious features appear and the model outputs  
 131 a correlated concept in open generation. We believe that extending to open-generation format

would require another layer of evaluation that captures the concepts/objects in generated outputs, either through LLM-as-a-judge or human annotations. As there are ongoing efforts in this layer of complexity and its potential biases, we leave this extension in format for future work.

## D Societal Impacts

While our work serves as a new spurious correlation benchmark for LVLMS, it is not to be used as a bullet-proof shield to claim that "a model with good accuracy on SpuriVerse can be free from all potential spurious correlation attacks," and thus reduce the efforts in improving the robustness of these models. As we also release the scene descriptions for the spurious group generation, malicious users can potentially design attacks more easily to generate harmful images while maintaining their usefulness to improve robustness against spurious correlation evaluated on SpuriVerse, causing a false promise of "better" models. In our work, we demonstrate the possibility of generalizing to unseen spurious correlations when finetuning on a diverse set of spurious correlations. We hope SpuriVerse can help foster more future work in this direction. We believe that SpuriVerse provides valuable insights into LVLMS' robustness to common spurious correlations when used properly.

## E Licenses

The anchor set of SpuriVerse is collected from AOKVQA [Schwenk et al., 2022], SEEDBench [Li et al., 2023], SEEDBench2 [Li et al., 2024b], NaturalBench [Li et al., 2024a].

The license or terms of use for each dataset and model is provided in the following:

Datasets:

AOKVQA: Apache-2.0.

SEEDBench: Attribution-NonCommercial 4.0 International.

SEEDBench2: Attribution-NonCommercial 4.0 International.

NaturalBench: Apache-2.0.

Models: GPT-4o: OpenAI's Term of Use and Business Terms

GPT-4o-mini: OpenAI's Term of Use and Business Terms

o4-mini: OpenAI's Term of Use and Business Terms

o3: OpenAI's Term of Use and Business Terms

Gemini 2.0 Flash: Google's API Terms of Service

Gemini 1.5 Pro: Google's API Terms of Service

Claude 3.7 Sonnet: Anthropic's Terms of Service

Qwen-VL-Max: Alibaba Cloud's Terms of Service

Qwen-VL-Pro: Alibaba Cloud's Terms of Service

Qwen2.5-vl-7b-instruct: Apache-2.0

Qwen2.5-vl-32b-instruct: Apache-2.0

Llama-3.2-11B-vision-instruct: Llama 3.2 Community License

Llama-3.2-90B-vision-instruct: Llama 3.2 Community License

LLaVA-v1.6 7b: LLAMA 2 Community License

LLaVA-v1.5: LLAMA 2 Community License

## References

- Anthropic. Claude: Ai assistant by anthropic, 2024. URL <https://www.anthropic.com>.
- Alibaba Cloud. Qwen api reference, 2025. URL <https://www.alibabacloud.com/help/en/model-studio/developer-reference/use-qwen-by-calling-api>.
- Michael Han Daniel Han and Unsloth team. Unsloth, 2023. URL <http://github.com/unslothai/unsloth>.
- Google DeepMind. Gemini 1.5 technical report, 2024. URL <https://deepmind.google/technologies/gemini/gemini-1-5>.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.



180 Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay  
181 Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural  
182 adversarial samples. *arXiv preprint arXiv:2410.14669*, 2024a.

183 Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking  
184 multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

185 Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench:  
186 Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer  
187 Vision and Pattern Recognition*, pages 13299–13308, 2024b.

188 Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning,  
189 2023a.

190 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural  
191 information processing systems*, 36:34892–34916, 2023b.

192 OpenAI. Gpt-4 technical report, 2023. URL <https://arxiv.org/abs/2303.08774>.

193 Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa:  
194 A benchmark for visual question answering using world knowledge. In *European conference on computer  
195 vision*, pages 146–162. Springer, 2022.

196 Qwen Team. Qwen2.5-vl, January 2025. URL <https://qwenlm.github.io/blog/qwen2.5-vl/>.

197 Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac,  
198 Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Brew, Alexis Nikolov, Sudhanshu Goyal, Yoann Dreyer,  
199 Julien Chaumond, and Alexander M. Rush. Transformers: State-of-the-art natural language processing.  
200 In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System  
201 Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi:  
202 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6>.