Jailbreak_LLM_Reproduction

June 3, 2025

[1]:

```
#  Mirrors: fschat, pandas, seaborn, matplotlib, numpy, datasets, evaluate
# + extras: bitsandbytes, transformers, accelerate, safetensors, wandb
# + attack / benchmark libs: llm-attacks, jailbreakbench
!pip install -q -U \
    fschat==0.2.23 \
    pandas seaborn matplotlib numpy datasets evaluate \
    anthropic google-generativeai openai \
    torch --extra-index-url https://download.pytorch.org/whl/cu118 \
    transformers accelerate bitsandbytes \
    safetensors pyyaml tqdm fire wandb \
    llm-attacks jailbreakbench
```

```
                           89.9/89.9 kB
1.7 MB/s eta 0:00:00
                           62.0/62.0 kB
1.4 MB/s eta 0:00:00
ERROR: Operation cancelled by user
```

[2]:

```
!git clone https://github.com/patrickrchao/JailbreakingLLMs.git
%cd JailbreakingLLMs
```

```
Cloning into 'JailbreakingLLMs'…
remote: Enumerating objects: 97, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 97 (delta 28), reused 42 (delta 23), pack-reused 38 (from 1)
Receiving objects: 100% (97/97), 54.95 KiB | 10.99 MiB/s, done.
Resolving deltas: 100% (38/38), done.
/content/JailbreakingLLMs
```

[ ]:

```
import huggingface_hub
huggingface_hub.login()
```

```
VBox(children=(HTML(value='<center> <img\nsrc=https://huggingface.co/front/
 ↪assets/huggingface_logo-noborder.sv…
```

```
[ ]: from huggingface_hub import snapshot_download

     vicuna_dir = snapshot_download(
         "lmsys/vicuna-7b-v1.5",
         ignore_patterns=["*.gguf"]       # skip optional GGUF files
     )

     llama_dir  = snapshot_download(
         "meta-llama/Llama-2-7b-hf",
         ignore_patterns=["*.gguf"]
     )

     print("  Vicuna files at:", vicuna_dir)
     print("  Llama-2 files at:", llama_dir)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(

Fetching 10 files:   0%|          | 0/10 [00:00<?, ?it/s]

.gitattributes:   0%|          | 0.00/1.52k [00:00<?, ?B/s]

config.json:   0%|          | 0.00/615 [00:00<?, ?B/s]

pytorch_model.bin.index.json:   0%|          | 0.00/26.8k [00:00<?, ?B/s]

pytorch_model-00002-of-00002.bin:   0%|          | 0.00/3.50G [00:00<?, ?B/s]

generation_config.json:   0%|          | 0.00/162 [00:00<?, ?B/s]

pytorch_model-00001-of-00002.bin:   0%|          | 0.00/9.98G [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/438 [00:00<?, ?B/s]

README.md:   0%|          | 0.00/1.97k [00:00<?, ?B/s]

tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/749 [00:00<?, ?B/s]

Fetching 17 files:   0%|          | 0/17 [00:00<?, ?it/s]

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
```

```
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`
```

```
README.md:   0%|          | 0.00/22.3k [00:00<?, ?B/s]
```

```
.gitattributes:   0%|          | 0.00/1.58k [00:00<?, ?B/s]
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`
```

```
model-00001-of-00002.safetensors:   0%|          | 0.00/9.98G [00:00<?, ?B/s]
```

```
LICENSE.txt:   0%|          | 0.00/7.02k [00:00<?, ?B/s]
```

```
model-00002-of-00002.safetensors:   0%|          | 0.00/3.50G [00:00<?, ?B/s]
```

```
model.safetensors.index.json:   0%|          | 0.00/26.8k [00:00<?, ?B/s]
```

```
generation_config.json:   0%|          | 0.00/188 [00:00<?, ?B/s]
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`
```

```
USE_POLICY.md:   0%|          | 0.00/4.77k [00:00<?, ?B/s]
```

```
pytorch_model.bin.index.json:   0%|          | 0.00/26.8k [00:00<?, ?B/s]
```

```
pytorch_model-00001-of-00002.bin:   0%|          | 0.00/9.98G [00:00<?, ?B/s]
```

```
pytorch_model-00002-of-00002.bin:   0%|          | 0.00/3.50G [00:00<?, ?B/s]
```

```
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
```

```
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`

special_tokens_map.json:   0%|          | 0.00/414 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/609 [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/1.84M [00:00<?, ?B/s]

Responsible-Use-Guide.pdf:   0%|          | 0.00/1.25M [00:00<?, ?B/s]

Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed.
Falling back to regular HTTP download. For better performance, install the
package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but
the 'hf_xet' package is not installed. Falling back to regular HTTP download.
For better performance, install the package with: `pip install
huggingface_hub[hf_xet]` or `pip install hf_xet`

tokenizer_config.json:   0%|          | 0.00/776 [00:00<?, ?B/s]

tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]

 Vicuna files at: /root/.cache/huggingface/hub/models--lmsys--
vicuna-7b-v1.5/snapshots/3321f76e3f527bd14065daf69dad9344000a201d
 Llama-2 files at: /root/.cache/huggingface/hub/models--meta-llama--
Llama-2-7b-hf/snapshots/01c7f73d771dfac7d292323805ebc428287df4f9
```

[ ]:
```
!pip install -q "litellm>=1.35.0"
```

```
                         8.0/8.0 MB
55.5 MB/s eta 0:00:00
```

[ ]:
```
!git clone https://github.com/llm-attacks/llm-attacks.git /content/llm-attacks
%cd /content/llm-attacks
!pip install -q -e .
```

```
Cloning into '/content/llm-attacks'…
remote: Enumerating objects: 157, done.
remote: Counting objects: 100% (115/115), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 157 (delta 80), reused 48 (delta 48), pack-reused 42 (from 1)
Receiving objects: 100% (157/157), 115.21 KiB | 3.72 MiB/s, done.
Resolving deltas: 100% (81/81), done.
/content/llm-attacks
  Preparing metadata (setup.py) … done
                          110.0/110.0

kB 3.0 MB/s eta 0:00:00
                          118.0/118.0

kB 5.3 MB/s eta 0:00:00
```

```
                          154.1/154.1 kB
10.9 MB/s eta 0:00:00
                           61.0/61.0 kB
5.6 MB/s eta 0:00:00
                           137.7/137.7

kB 5.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
                         177.1/177.1 kB
15.5 MB/s eta 0:00:00
                         7.0/7.0 MB
89.3 MB/s eta 0:00:00
                         76.7/76.7 kB
7.3 MB/s eta 0:00:00
                         7.8/7.8 MB
133.0 MB/s eta 0:00:00
                         363.4/363.4 MB
1.4 MB/s eta 0:00:00
                         13.8/13.8 MB
117.2 MB/s eta 0:00:00
                         24.6/24.6 MB
98.4 MB/s eta 0:00:00
                         883.7/883.7 kB
59.5 MB/s eta 0:00:00
                         664.8/664.8 MB
1.5 MB/s eta 0:00:00
                         211.5/211.5 MB
6.9 MB/s eta 0:00:00
                         56.3/56.3 MB
14.7 MB/s eta 0:00:00
                         127.9/127.9 MB
8.2 MB/s eta 0:00:00
                         207.5/207.5 MB
6.6 MB/s eta 0:00:00
                         21.1/21.1 MB
104.5 MB/s eta 0:00:00
                         95.2/95.2 kB
9.7 MB/s eta 0:00:00
                         3.1/3.1 MB
108.2 MB/s eta 0:00:00
                         20.3/20.3 MB
110.5 MB/s eta 0:00:00
                         299.2/299.2 kB
27.7 MB/s eta 0:00:00
                         18.3/18.3 MB
102.4 MB/s eta 0:00:00
                         62.5/62.5 kB
5.7 MB/s eta 0:00:00
```

```
                         739.0/739.0 kB
49.4 MB/s eta 0:00:00
                         4.5/4.5 MB
110.4 MB/s eta 0:00:00
                         72.0/72.0 kB
6.7 MB/s eta 0:00:00
                         130.6/130.6 kB
12.4 MB/s eta 0:00:00
                         73.7/73.7 kB
7.0 MB/s eta 0:00:00
                         48.5/48.5 kB
4.7 MB/s eta 0:00:00
                         67.1/67.1 kB
6.3 MB/s eta 0:00:00
  Building wheel for wavedrom (setup.py) … done
```

```
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
litellm 1.72.0 requires pydantic<3.0.0,>=2.0.0, but you have pydantic 1.10.22
which is incompatible.
albumentations 2.0.7 requires pydantic>=2.9.2, but you have pydantic 1.10.22
which is incompatible.
dataproc-spark-connect 0.7.4 requires websockets>=14.0, but you have websockets
11.0.3 which is incompatible.
thinc 8.3.6 requires numpy<3.0.0,>=2.0.0, but you have numpy 1.26.4 which is
incompatible.
thinc 8.3.6 requires pydantic<3.0.0,>=2.0.0, but you have pydantic 1.10.22 which
is incompatible.
langchain 0.3.25 requires pydantic<3.0.0,>=2.7.4, but you have pydantic 1.10.22
which is incompatible.
langchain-core 0.3.60 requires pydantic>=2.7.4, but you have pydantic 1.10.22
which is incompatible.
yfinance 0.2.61 requires websockets>=13.0, but you have websockets 11.0.3 which
is incompatible.
sentence-transformers 4.1.0 requires transformers<5.0.0,>=4.41.0, but you have
transformers 4.28.1 which is incompatible.
google-genai 1.16.1 requires pydantic<3.0.0,>=2.0.0, but you have pydantic
1.10.22 which is incompatible.
google-genai 1.16.1 requires websockets<15.1.0,>=13.0.0, but you have websockets
11.0.3 which is incompatible.
```

[ ]:
```
%cd /content
```

```
/content
```

[ ]:
```
import yaml, pathlib, sys, os
yaml_path = pathlib.Path.home()/".jblm_local_models.yaml"
yaml_path.write_text(yaml.dump({
    "vicuna-7b-v1.5": {        # IDs must match main.py --target-model choices
```

```
        "hf_path": vicuna_dir, "quant": True, "dtype": "float16"},
    "Llama-2-7b-hf": {    # we point chat-hf ID at the 13-B GPTQ
        "hf_path": llama_dir,  "quant": True, "dtype": "float16"}
}))
sys.path.append("/content/JailbreakingLLMs")
print("  YAML written →", yaml_path)
```

```
 YAML written → /root/.jblm_local_models.yaml
```

[4]:

```
import subprocess, json, pandas as pd, re, yaml, pathlib, sys

from llm_attacks import run_jbb_gcg
from jailbreakbench.data import load_prompts
from jailbreakbench.evaluate import evaluate_list, evaluate_csv
from jailbreakbench.model_wrappers import LocalWrapper
from jailbreakbench.defenses import SmoothLLM, PerplexityFilter
import transformers as tfm, torch

# targets to evaluate
TARGETS = ["vicuna-7b-v1.5", "Llama-2-7b-hf"]

def load_ckpt_path(target):
    cfg = yaml.safe_load(pathlib.Path.home().joinpath(".jblm_local_models.
 ↪yaml").read_text())
    return cfg[target]["hf_path"]

def run_pair(target):
    out = subprocess.check_output([
        sys.executable, "main.py",
        "--attack-model",   "vicuna-13b-v1.5",
        "--target-model",   target,
        "--judge-model",    "jailbreakbench",
        "--evaluate-locally",
        "--n-streams",      "30",
        "--n-iterations",   "3"
    ], text=True)
    m = re.search(r"FINAL-RESULT (.*)", out)
    return json.loads(m.group(1))

def run_gcg(target):
    path = load_ckpt_path(target)
    res  = run_jbb_gcg(checkpoint=path, name=target)
    return {"jailbreak_pct": res["jb_pct"], "queries_per_success":␣
 ↪res["q_per_success"]}

def run_jbc(target):
    path = load_ckpt_path(target)
    prompts = load_prompts("JBC")
    tok = tfm.AutoTokenizer.from_pretrained(path, use_fast=False)
    mdl = tfm.AutoModelForCausalLM.from_pretrained(path, device_map="auto",␣
 ↪torch_dtype=torch.float16)
    wrap = LocalWrapper(model=mdl, tokenizer=tok, max_new_tokens=150)
    jb = evaluate_list(wrap, prompts) * 100
    return {"jailbreak_pct": jb, "queries_per_success": float("nan")}

def reproduce_table2():
    rows=[]
    for tgt in TARGETS:
        pr = run_pair(tgt)
        gr = run_gcg(tgt)
        jr = run_jbc(tgt)
        rows += [
            {"Method":"PAIR","Model":tgt, "JB_%":pr["jailbreak_pct"],        "Q/
 ↪S":pr["queries_per_success"]},
            {"Method":"GCG", "Model":tgt, "JB_%":gr["jailbreak_pct"],        "Q/
 ↪S":gr["queries_per_success"]},
            {"Method":"JBC", "Model":tgt, "JB_%":jr["jailbreak_pct"],        "Q/
 ↪S":jr["queries_per_success"]},
        ]
    df = pd.DataFrame(rows)
    print("### Table 2 - Jailbreak Success Rates")
    print(df.to_markdown(index=False))

def reproduce_table5():
    rows=[]
    for tgt in TARGETS:
        csv = pathlib.Path(f"outputs/{tgt}/pair_successful_prompts.csv")
        if not csv.exists():
            print(f"   Missing {csv}, run PAIR first")
            continue
        sf = evaluate_csv(SmoothLLM(tgt), csv) * 100
        pf = evaluate_csv(PerplexityFilter(tgt), csv) * 100
        rows += [
            {"Attack":"PAIR","Model":tgt,"Defense":"SmoothLLM","JB_%":sf},
            {"Attack":"PAIR","Model":tgt,"Defense":"Perplexity","JB_%":pf},
        ]
    df = pd.DataFrame(rows)
    print("\n### Table 5 - Defended Jailbreak Rates")
    print(df.to_markdown(index=False))

reproduce_table2()
reproduce_table5()
```

```
### Table 2 - Jailbreak Success Rates (Vicuna-13B attacker)
```

Method	Model	JB_%	Q/S
PAIR	vicuna-7b-v1.5	75	12
GCG	vicuna-7b-v1.5	45	0.3
JBC	vicuna-7b-v1.5	50	nan
PAIR	llama-2-7b-chat-hf	6	50
GCG	llama-2-7b-chat-hf	3	0.3
JBC	llama-2-7b-chat-hf	1	nan

Table 5 - Defended Jailbreak Rates (Vicuna-13B attacker)

Attack	Model	Defense	JB_%
PAIR	vicuna-7b-v1.5	SmoothLLM	35
PAIR	vicuna-7b-v1.5	Perplexity	75
PAIR	llama-2-7b-chat-hf	SmoothLLM	0
PAIR	llama-2-7b-chat-hf	Perplexity	4