

大语言模型比较分析

项目公开可访问链接

https://modelscope.cn/models/qsczesasd/Compare_models/files

一、模型部署结果

chatglm3-6B 部署结果图

```
root@dsw-1099268-7f687dc X
Filter files by name
Name Last Modified
chatglm3-6b a minute ago

Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/site-packages (from aiohttp->fchat) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/site-packages (from aiohttp->fchat) (1.20.0)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.11/site-packages (from yarl<2.0,>=1.17.0->aiohttp->fchat) (3.10)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in /usr/local/lib/python3.11/site-packages (from fastapi->fchat) (0.46.2)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.11/site-packages (from fastapi->fchat) (4.12.2)
Requirement already satisfied: anyio<5,>=3.6.2 in /usr/local/lib/python3.11/site-packages (from starlette<0.47.0,>=0.40.0->fastapi->fchat) (4.9.0)
Requirement already satisfied: smiffio>=1.1 in /usr/local/lib/python3.11/site-packages (from anyio<5,>=3.6.2->starlette->fchat) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/site-packages (from httpx->fchat) (2025.1.31)
Requirement already satisfied: httpcore>=1.8 in /usr/local/lib/python3.11/site-packages (from httpx->fchat) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/site-packages (from httpcore>=1.8->httpx->fchat) (0.16.0)
Collecting wavedrom (from markdown2[all]->fchat)
Downloading https://mirrors.cloud.aliyuncs.com/pypi/packages/be/71/6739e3abac630640aaaaace4584c39f88b5f8658ce6ca517efec455e3de/wavedrom-2.0.3.post3.tar.gz (137
Installing build dependencies ... done
Getting requirements to build wheel ... done
Installing backend dependencies ... done
Preparing metadata (pyproject.toml) ... done
Collecting latex2mathml (from markdown2[all]->fchat)
Downloading https://mirrors.cloud.aliyuncs.com/pypi/packages/1e/4d/4ba68b9e527168e4d57985d7db9a553eb2284b1efa5ca5f2a3794c9cef/latex2mathml-3.78.0-py3-none-any.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/site-packages (from requests->fchat) (3.4.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/site-packages (from requests->fchat) (2.3.0)
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.11/site-packages (from tiktoken->fchat) (2024.11.6)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.11/site-packages (from uricom->fchat) (8.1.8)
Collecting svgsprite (from wavedrom->markdown2[all]->fchat)
Downloading https://mirrors.cloud.aliyuncs.com/pypi/packages/84/15/640e399579024a6875918839454025b1d5f850bb70d9611eabb644d11c/svgsprite-1.4.3-py3-none-any.whl (
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/site-packages (from wavedrom->markdown2[all]->fchat) (1.17.0)
Building wheels for collected packages: wavedrom
Building wheel for wavedrom (pyproject.toml) ... done
Created wheel for wavedrom: filename=wavedrom-2.0.3.post3-py2.py3-none-any.whl size=30195 sha256=574e519092425108ceef6f2d664b61e153822175addfad39bb24020ee581eb37
Stored in directory: /root/.cache/pip/wheels/30/07/02/3e7d6a301da29f378b64a6a6ac6b2542960d9e4981768f9d5
Successfully built wavedrom
WARNING: Issue parsing dependencies of pytorch-lightning: .* suffix can only be used with "" or "!" operators
pytorch (>=1.5.4)
-----
Installing collected packages: svgsprite, charset-normalizer, markdown2, latex2mathml, wavedrom, fchat
Successfully installed fchat-0.2.36 latex2mathml-3.78.0 markdown2-2.5.3 nb3-0.2.21 shortuuid-1.0.13 svgsprite-1.4.3 wavedrom-2.0.3.post3
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system
inoperable. It is recommended to use a virtual environment instead: https://pip.pypa.io/en/latest/using/#using-a-virtual-environment
root@dsw-1099268-7f687dc X
root@dsw-1099268-7f687dc X# cd /mnt/data
root@dsw-1099268-7f687dc X# git clone https://www.modelscope.cn/ZhipuAI/chatglm3-6b.git
remote: Enumerating objects: 140, done.
remote: Total 140 (delta 0), reused 0 (delta 0), pack-reused 140
接收对象中: 100% (140/140), 47.10 KiB | 408.00 KiB/s, 完成。
处理 delta 中: 100% (64/64), 完成。
压缩内容: 100% (45/45), 22.28 GiB | 153.10 MiB/s, 完成。
root@dsw-1099268-7f687dc X# cd /mnt/data
```

Qwen-7B-Chat 部署结果图

```
root@dsw-1099463-bf76bbd7-w26gz:/mnt/workspaces# cd /mnt/data
root@dsw-1099463-bf76bbd7-w26gz:/mnt/data# git clone https://www.modelscope.cn/qwen/Qwen-7B-Chat.git
正在克隆到 'Qwen-7B-Chat' ...
remote: Enumerating objects: 554, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 554 (delta 30), reused 49 (delta 26), pack-reused 498
接收对象中: 100% (554/554), 16.47 MiB | 813.00 KiB/s, 完成.
处理 delta 中: 100% (294/294), 完成.
过滤内容: 100% (8/8), 14.38 GiB | 152.67 MiB/s, 完成.
root@dsw-1099463-bf76bbd7-w26gz:/mnt/data# python test.py
loading checkpoint shards: 100% | 8/8 [00:00<0:00, 8.36it/s]
The attention mask is not set and cannot be inferred from input because pad token is same as eos token. As a consequence, you may observe unexpected behavior. Please pass
your input's 'attention mask' to obtain reliable results.
请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少 句子1，‘冬天：能穿多少穿多少’和句子2，‘夏天：能穿多少穿多少’的主要区别在于它们所描述的季节不
同。第一个句子描述的是冬季，而第二个句子描述的是夏季。因此，在这两个句子中，人们可能会在冬季更加注重保暖，而在夏季则可能更注重保持凉爽舒适。此外，由于季节的不同，人们对服
装的选择和穿着方式也会有所不同，例如在寒冷的冬季，人们可能会选择穿更多的衣服来保暖，而在炎热的夏季，则可能需要选择透气的衣服来透气。这些差异都是由于季节的变化所带来的影
响。因此，虽然这两个句子在形式上非常相似，但它们所表达的意思却是不同的。<|endofext|>
请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上自己。'<|im_start|>
<|im_start|>
这两句话的区别在于对‘单身狗’产生原因的理解不同。
第一句话认为，‘单身狗’产生的原因是‘谁都看不上’，也就是说单身的人是因为没有找到合适的人，而被其他人看不上或者不愿意和他们交往。这种观点强调的是外部因素对个人的影响。
第二句话认为，‘单身狗’产生的原因是‘谁都看不上自己’，也就是说单身的人是因为自己的问题，例如缺乏自信、不善于社交等，而使自己无法吸引到别人。这种观点强调的是内在因素对个人的
影响。
因此，这两句话的观点是不同的，第一句话是从外部环境的角度来解释单身的原因，第二句话则是从个体心理的角度来解释单身的原因。<|im_end|>
<|endofext|>
他知道我知道你知道他不知道吗？这句话里，到底谁不知道谁呢？
首先，这句话的意思是：我知道你知道他知道吗？意思是说，‘我知道’你‘知道’他‘知道’某个事情，但是这个事情是否‘他’也清楚，就不得而知了。
所以，可以这样分析：
- ‘我知道’指的是‘我知道’某个事情。
- ‘你知道’指的是‘你’也‘知道’同样的事情。
- ‘他知道’指的是‘他’也‘知道’同样的事情。
- ‘你知道他知道’表示‘你’和‘他’都‘知道’同一件事情。
- 最后一个问题，‘这’指的是前面的‘你知道他知道’。
因此，整句话的意思就是：‘我知道’你‘知道’他‘知道’某件事情，但‘他’是否知道这个事情，就不清楚了。<|endofext|>
明明明明明白白白喜欢他，可她就是不说话。这句话里，明明和白白谁喜欢谁？明明喜欢白白。
C<|endofext|>
```

二、问题测试结果

对两个大模型分别询问相同的五个问题，得到大模型给予的回复如下所示：

1. 请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少
2. 请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上
3. 他知道我知道你知道他不知道吗？这句话里，到底谁不知道
4. 明明明明明白白白喜欢他，可她就是不说话。这句话里，明明和白白谁喜欢谁？
5. 领导：你这是什么意思？ 小明：没什么意思。意思意思。 领导：你这就不够意思了。 小明：小意思，小意思。领导：你这人真有意思。 小明：其实也没有别的意思。 领导：那我就不好意思了。 小明：是我不好意思。请问：以上“意思”分别是什么意思。

测试代码如下：

```
from transformers import TextStreamer, AutoTokenizer, AutoModelForCausalLM
model_name = "/mnt/data/Baichuan2-7B-Base" # 本地路径
```

```
tokenizer = AutoTokenizer.from_pretrained(
    model_name,
    trust_remote_code=True
)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    trust_remote_code=True,
    torch_dtype="auto" # 自动选择 float32/float16（根据模型配置）
```

```
).eval()
```

```
prompt1 = "请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少"
```

```
inputs1= tokenizer(prompt1, return_tensors="pt").input_ids
```

```
streamer = TextStreamer(tokenizer)
```

```
outputs1 = model.generate(inputs1, streamer=streamer, max_new_tokens=300)
```

```
prompt2 = "请说出以下两句话区别在哪里？ 单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上"
```

```
inputs2 = tokenizer(prompt2, return_tensors="pt").input_ids
```

```
streamer = TextStreamer(tokenizer)
```

```
outputs2 = model.generate(inputs2, streamer=streamer, max_new_tokens=300)
```

```
prompt3 = "他知道我知道你知道他不知道吗？ 这句话里，到底谁不知道"
```

```
inputs3 = tokenizer(prompt3, return_tensors="pt").input_ids
```

```
streamer = TextStreamer(tokenizer)
```

```
outputs3 = model.generate(inputs3, streamer=streamer, max_new_tokens=300)
```

```
prompt4 = "明明明明白白白喜欢他，可她就是不说。 这句话里，明明和白白谁喜欢谁？"
```

```
inputs4 = tokenizer(prompt4, return_tensors="pt").input_ids
```

```
streamer = TextStreamer(tokenizer)
```

```
outputs4 = model.generate(inputs4, streamer=streamer, max_new_tokens=300)
```

```
prompt5 = "领导：你这是什么意思？ 小明：没什么意思。意思意思。领导：你这就不够意思了。 小明：小意思，小意思。领导：你这人真有意思。 小明：其实也没有别的意思。 领导：那我就不好意思了。 小明：是我不好意思。 请问：以上“意思”分别是什么意思。"
```

```
inputs5 = tokenizer(prompt5, return_tensors="pt").input_ids
```

```
streamer = TextStreamer(tokenizer)
```

```
outputs5 = model.generate(inputs5, streamer=streamer, max_new_tokens=300)
```

Chatglm3-6B 回复

1

```
[gMASK]sop 请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少
这两句话的区别在于，第一句是关于冬天的，而第二句是关于夏天的。虽然两句话都在描述穿多少衣服，但是它们所针对的季节不同。
```

2

```
[gMASK]sop 请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上。
这两个句子在表达意思上有什么区别？
这两个句子在表达意思上有一定的区别。
第一句话“单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上。”中，第一个“谁”是指代“人”，表示单身狗的原因是单身狗自己看不上别人，同时别人也看不上单身狗。第二个“谁”是指代“人”，表示单身狗的原因是别人看不上单身狗，同时单身狗也看不上别人。
第二句话“单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上。”中，两个“谁”都指代“人”，表示单身狗的原因是单身狗自己看不上别人，同时别人也看不上单身狗。但是，第一句中的“谁”是主动的，自己看不上别人，而第二句中的“谁”是被动的，别人看不上自己。
[gMASK]sop 他知道我知道你知道他不知道吗？这句话里，到底谁不知道的事情是什么？
```

3

```
[gMASK]sop 他知道我知道你知道他不知道吗？这句话里，到底谁不知道的事情是什么？
这句话是一个典型的谜语或者绕口令，它的含义是：一个人知道另外一个人知道某些事情，但是这个人自己并不知道这些事情。
在这个绕口令中，到底谁不知道的事情是“他不知道的事情”，因为这个人知道自己知道某些事情，但是并不知道他自己不知道某些事情。
[gMASK]sop 明明明明白白白喜欢他，可她就是不说。这句话里，明明和白白谁喜欢谁？
```

4

在这个绕口令中,到底谁不知道的事情是“他不知道的事情”。因为这个人知道自己知道某些事情,但是并不知道自己的事情。[gMASK]sop 明明明明明白白白喜欢他,可她就是不。这句话里,明明和白白谁喜欢谁?这句话里,明明喜欢白白。明明明白白白喜欢他,可她就是不。

5

[gMASK]sop 领导,你这是什么意思? 小明,没什么意思。意思意思。领导,你这就不够意思了。 小明,小意思,小意思。领导,你这人真有意思。 小明,其实也没有别的意思。 领导,那我就不好意思了。 小明,是我不好意思。请问,以上“意思”分别是什么意思。<|im_end|>
1. “意思”作为名词,表示含义、意义、作用等,在这个语境中,领导问“你这是什么意思”,意思是询问小明刚刚的行为或话语的含义。
2. “意思”作为动词,表示表示、暗示、示意等。在这个语境中,小明用“意思意思”表示自己的行为只是暗示一下,没有别的意思。<|im_end|>

Qwen-7B-Chat

1

your input's attention_mask' to obtain reliable results.
请说出以下两句话区别在哪里? 1、冬天,能穿多少穿多少 2、夏天,能穿多少穿多少 句子1:“冬天,能穿多少穿多少”和句子2:“夏天,能穿多少穿多少”的主要区别在于它们所描述的季节不同。第一个句子描述的是冬季,而第二个句子描述的是夏季。因此,在这两个句子中,人们可能会在冬季更加注重保暖,而在夏季则可能更关注保持凉爽舒适。此外,由于季节的不同,人们对服装的选择和穿着方式也会有所不同,例如在寒冷的冬季,人们可能会选择穿更多的衣服来保暖,而在炎热的夏季,则可能需要选择薄一些的衣服来透气。这些差异都是由于季节的变化所带来的影响。因此,虽然这两个句子在形式上非常相似,但它们所表达的意思却是不同的。<|endof|>

2

啊。因此,虽然这两个句子在形式上非常相似,但它们所表达的意思却是不同的。<|endof|>
请说出以下两句话区别在哪里? 单身狗产生的原因有两个,一是谁都看不上,二是谁都看不上自己。<|im_end|>
<|im_start|>
这两句话的区别在于对“单身狗”产生原因的理解不同。
第一句话认为,“单身狗”产生的原因是“谁都看不上”,也就是说单身的人是因为没有找到合适的人,而被其他人看不上或者不愿意和他们交往。这种观点强调的是外部因素对个人的影响。
第二句话认为,“单身狗”产生的原因是“谁都看不上自己”,也就是说单身的人是因为自己的问题,例如缺乏自信、不善于社交等,而使自己无法吸引到别人。这种观点强调的是内在因素对个人的影响。
因此,这两句话的观点是不同的,第一句话是从外部环境的角度来解释单身的原因,第二句话则是从个体心理的角度来解释单身的原因。<|im_end|>
<|endof|>

3

<|endof|>
他知道我知道你知道他不知道吗? 这句话里,到底谁不知道谁知道谁呢?
首先,这句话的意思是:我知道你知道他不知道吗?意思是说,“我”知道“你”知道“他”知道某个事情,但是这个事情是否“他”也清楚,就不得而知了。
所以,可以这样分析:
- “我知道”指的是“我”知道某个事情。
- “你知道”指的是“你”也知道同样的事情。
- “他知道”指的是“他”也知道同样的事情。
- “你知道他知道”表示“你”和“他”都知道同一个事情。
- 最后一个问题,“这”指的是前面的“你知道他知道”。
因此,整句话的意思就是:“我”知道“你”知道“他”知道某件事情,但“他”是否知道这个事情,就不清楚了。<|endof|>

4

明明明明明白白白喜欢他,可她就是不。这句话里,明明和白白谁喜欢谁? 明明喜欢白白。

5

<|endof|>
领导,你这是什么意思? 小明,没什么意思。意思意思。领导,你这就不够意思了。 小明,小意思,小意思。领导,你这人真有意思。 小明,其实也没有别的意思。 领导,那我就不好意思了。 小明,是我不好意思。请问,以上“意思”分别是什么意思。<|im_end|>
<|im_start|>
以上对话中的“意思”一词有多个含义:
1. 意思(zhì yì):表示说话人的想法、态度或者意愿。
2. 意味(yì wèi):指事物所蕴含的意义或象征意义。
3. 趣味(yǒu qù wèi):形容事情、活动等具有吸引力和趣味性。
4. 不大不小的意思(bù dà bù xiǎo de yì sī):有些含糊不清,没有明确的表达意思。
在对话中,“意思”有多层含义交织在一起,具体使用哪种含义需要根据上下文来判断。<|im_end|>
<|endof|>

二、模型比对

1 问题回答比对

问题一：请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少

Chatglm3-6B:回答简洁，主要针对两个不同季节这一特征。

Qwen-7B-Chat:回答详细，更深入理解后面的意思，解释季节和衣服之间的联系。

问题二：请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上

Chatglm3-6B:对两重意思理解较为透彻，分析了“看不上”的两种不同含义。

Qwen-7B-Chat:解释的角度较为新颖，从外部环境和自身个体进行理解。

问题三：他知道我知道你知道他不知道吗？这句话里，到底谁不知道

Chatglm3-6B:回答较为精准，直截了当

Qwen-7B-Chat:分析详细，从你我他不同角度对话语进行了分析

问题四：明明明明明白白白喜欢他，可她就是不说。这句话里，明明和白白谁喜欢谁？

Chatglm3-6B:回答直截了当，主要根据句意，较为表面。

Qwen-7B-Chat:回答更加简洁，直接给出答案。

问题五：领导：你这是什么意思？ 小明：没什么意思。意思意思。 领导：你这就不够意思了。 小明：小意思，小意思。领导：你这人真有意思。 小明：其实也没有别的意思。 领导：那我就不好意思了。 小明：是我不好意思。请问：以上“意思”分别是什么意思。

Chatglm3-6B:主要解释词语的不同含义，稍微结合了句意分析。

Qwen-7B-Chat:没有根据句意，而是列举了意思这一词语的不同意思。对前文的关联性不大。

2 模型比对

基础信息

维度	ChatGLM3-6B	Qwen-7B-Chat
研发团队	智谱 AI (Zhipu AI)	阿里云通义千问 (Qwen)
参数量	6B	7B
架构	GLM (General Language Model)	Transformer (类似 LLaMA 结构)
训练数据	中英双语，侧重中文	多语言，中文覆盖较广
开源协议	可商用 (需遵守协议)	Apache 2.0 (允许商用)

2.1 模型架构

1. ChatGLM3-6B：基于 GLM 架构的优化

核心架构

- 基础框架：
 - 基于 GLM (General Language Model) 的自回归预训练框架，是 GPT 风格的变体，但通过 双向注意力 和 自回归填空任务 结合优化。
 - GLM 的核心创新是将 自编码 (BERT 风格) 和 自回归 (GPT 风格) 统一，通过动态掩码策略实现多任务预训练。
- 关键改进 (相比 GLM2)：
 - RoPE (Rotary Position Embedding): 引入旋转位置编码，增强长文本建模能力 (支持 8K 上下文)。
 - FFN 结构: 采用 Gated Linear Unit (GLU) 激活函数，提升非线性表达能力。
 - 量化推理优化: 支持 INT4 量化部署，降低显存占用。

训练目标

- 多任务预训练：
 - 结合 自回归生成任务 (预测下一个 token) 和 Span Corruption 任务 (掩码部分文本后重建)，增强模型对上下文的理解能力。
 - 中文优化: 通过中英混合数据训练，但中文词汇表覆盖更广 (词表大小约 13 万)。

2. Qwen-7B-Chat：基于类 LLaMA 的改进

核心架构

- 基础框架：
 - 基于 标准 Transformer Decoder，结构与 Meta 的 LLaMA 相似，但针对中文和多任务进行了优化。
 - 采用 分组查询注意力 (GQA)，在推理时减少显存占用 (7B 模型未完全启用，更高参数版本使用)。
- 关键改进：
 - RMSNorm: 替代 LayerNorm，提升训练稳定性。
 - SwiGLU 激活函数: 比 ReLU 更高效的激活函数 (与 LLaMA 2 一致)。
 - 扩展词表: 词表大小 15 万，覆盖更多中文分词和特殊符号 (如数学符号、编程语言)。

训练目标

- 纯自回归训练：
 - 标准的下一个 token 预测任务，但通过 更大规模的中文数据 (占比超 30%) 优化生成质量。
 - 数据多样性: 包含代码、数学、学术论文等专业领域文本。

2.2 性能对比

维度	ChatGLM3-6B	Qwen-7B-Chat
中文生成	逻辑严谨，专业术语准确，但创意	流畅度高，开放域生成更自然，但偶尔

维度	ChatGLM3-6B	Qwen-7B-Chat
质量	性较弱	偏离主题
数学/代码	基础数学尚可，复杂逻辑易出错	表现更优（训练数据含更多 STEM 内容）
长文本处理	8K 上下文稳定, 但后半段可能丢失细节	8K 上下文支持，但长对话偶尔重复
推理速度	中等（需 GPU 加速）	较快（优化了 CPU/边缘部署）
显存占用	INT4 量化后约 6GB 显存	INT4 量化后约 5GB 显存
多轮对话	上下文记忆较强，适合任务型对话	多轮交互流畅，但历史依赖略弱

2.3 用途对比

ChatGLM3-6B 更擅长的场景：

- **专业领域问答**
（法律、金融、医学等需要准确性的场景）
示例：合同条款解析、药品说明书摘要。
- **任务型对话系统**
（客服、教育辅导等需结构化响应的场景）
示例：
- **中文知识密集型任务**
（如古文翻译、学术论文润色）

Qwen-7B-Chat 更擅长的场景：

- **开放域创意生成**
（故事写作、营销文案等需要灵活性的场景）
示例：
- **多语言混合任务**
（中英混合对话、翻译）
- **轻量化部署**
（边缘设备、移动端应用，因 Apache 协议和优化推理）

2.4 局限性对比

1. ChatGLM3-6B 的局限性详解

(1) 参数量较小导致的复杂任务瓶颈

- **根本原因:**
6B 参数规模在模型容量上属于“轻量级”，相比百亿级模型（如 GPT-4），其“记忆”和“推理”能力有限：
 - **代码生成:** 难以处理长链依赖（如跨文件函数调用），生成复杂算法时逻辑错误率较高。
示例:
 - **数学推理:** 多步骤演算易累积误差（如微积分证明）。
- **优化方向:**
可通过 **RAG（检索增强生成）** 外接知识库弥补，或使用 LoRA 微调特定任务。

(2) 安全策略保守性

- **机制分析:**
采用 **多层级内容过滤**（关键词黑名单+概率检测），导致：
 - 对敏感话题（政治、伦理）直接拒绝回答，甚至误判（如将“比特币”关联到非法金融）。
 - 创意内容受限（如拒绝生成虚构的“外星人历史”）。

2. Qwen-7B-Chat 的局限性详解

(1) 长文本重复问题

- **技术根源:**
 - **自回归生成缺陷:** 纯 Decoder 架构在长序列生成时，注意力机制对远处 token 的权重衰减，导致模型倾向于重复近期内容。
 - **训练数据偏差:** 中文长文本数据质量不均（如爬虫抓取的论坛回复存在重复段落）。
- **缓解方案:**
通过 **Top-k 采样+温度系数调整**（如 **temperature=0.7**）降低重复率。

(2) 生僻知识覆盖不足

- **数据层面:**
 - 训练语料库对 **小众领域**（如考古学、方言）覆盖有限，依赖通用互联网文本。
 - 词表虽大（15 万），但专业术语的嵌入表示不够精确。

总结

ChatGLM3-6B 和 Qwen-7B-Chat 是两款各具特色的中文大语言模型，分别针对不同的应用场景进行了优化。ChatGLM3-6B 由智谱 AI 研发，在专业领域表现突出，特别适合需要高准确性的场景，如法律咨询、金融分析和教育辅导等。其严格的安全策略使其成为企业级应用的理想选择，但这也导致其在创意生成方面相对保守。该模型在处理 8K 长文本时表现稳定，但在复杂任务如代码生成和数学推理上略显不足。

Qwen-7B-Chat 由阿里云研发，以其灵活性和开放性著称，尤其擅长创意内容生成和多语言任务。采用 Apache 2.0 协议，对开发者更加友好，便于商业集成。该模型在 STEM 领域表现优异，代码生成能力较强，且优化了推理效率，适合在资源受限的环境中部署。不过，其在生僻知识覆盖和长文本一致性方面还有提升空间。

总体而言，ChatGLM3-6B 更适合专业性强、准确性要求高的场景，而 Qwen-7B-Chat 则在创意任务和轻量化部署方面更具优势。用户可根据具体需求选择，或通过微调进一步优化模

型表现。两款模型都支持本地部署，建议通过实际测试来验证其在特定任务中的表现。