为什么现在大模型在推理阶段都是左padding?

复盘一下。

如果再被问一次，我会这样回答：

> 🤖 decoder-only 大模型在**推理**时通常选左padding。原因是生成时模型默认用序列最后一个 token 的 logits 采样下一步。如果右 padding，最后一个位置是 <pad>，采样逻辑会拿到无意义的 logits；而左 padding 保证最后一个位置永远是真实 token。
>
> 此外左 padding 让不同长度句子右对齐，KV-cache 能复用，在线推理显存和吞吐都更高效。
>
> **训练**的时候左padding还是右padding没有影响，因为可以自行设置ignore_label忽略掉padding的位置，但训练资源紧张，一般也不padding，直接constant length data loader。

什么叫padding?

- **在训练和推理中，为了批量并行处理可变长度的序列，需要对短序列进行填充。**
- **做法**：往句子里塞“占位符 token（`<pad>`）”。
  - **左padding**：`<pad> <pad> 我 喜欢 苹果`
  - **右padding**：`我 喜欢 苹果 <pad> <pad>`

为啥“大模型”几乎都用左 padding?

1、生成时只看“最后一个”logit

generate()的过程中需要从最后一个token的probability中sample下一个token，但right padding时最后一个token是<pad>，模型就会用 **占位符的 logits 来采样**，直接翻车。HF Transformers 因此会警告

“A decoder-only architecture is being used, but right-padding was detected! For correct generation results, please set `padding_side='left'` when initializing the tokenizer.”

2、KV-cache

在线服务时会把已算好的 Key/Value 缓存住。左 padding 让“真正的 token”在每条序列的 **右端对齐**，这样批里不同长度请求共用同一块 KV-cache，GPU 读写整齐，像 vLLM 的 paged-attention 就是这么设计的。

那右 padding 就一无是处吗?

- **BERT/encoder-only** 这种一次性读完再做分类的模型，输出不依赖“最后一个位置”，左右 padding 都行，很多教程还是右 padding。
- **训练阶段切固定长度块**（pre-training “pack” trick）：干脆不用 `<pad>`。所以论文里常见“我们没用 padding”。

实战小贴士

代码块

```
tok = AutoTokenizer.from_pretrained("meta-llama/Llama-3-8B")
tok.pad_token = tok.eos_token        # Llama 没有原生 pad
tok.padding_side = "left"            # 关键行
out = tok(texts, padding=True, return_tensors='pt')
```