ChatGPT

Context and Example Configuration

In the original Axolotl example for **Llama-3.1-SuperNova-Lite-Reflection-V1.0** (Axolotl v0.4.1), the dataset was specified using the old `sharegpt` format. The relevant config snippet was:

```
datasets:
  - path: SE6446/MAGllama_Sharegpt
    type: sharegpt
    conversation: chatml
```

This shows the `sharegpt` dataset type (now deprecated) with a ChatML conversation template [1]. The underlying dataset **SE6446/MAGllama_Sharegpt** is a ShareGPT-style conversational dataset: each example has a `conversations` field that is a list of messages with a `"from"` and `"value"` (role and content) [2]. (This dataset schema matches the ShareGPT JSON format.)

> **Dataset format:** Each row has
>
> ```
> {"conversations":[{"from":"user","value":"..."},
> {"from":"assistant","value":"..."}]}
> ```
>
> e.g. see the dataset features listing in its README [2].

Because `type: sharegpt` is now **deprecated** (modern Axolotl uses `type: chat_template` instead) [3] [4], new configs replace it with a chat template format (e.g. `type: chat_template` with `field_messages: conversations` and `message_property_mappings`). However, the old example uses `sharegpt` and the ChatML template.

Model Tokenizer Expectations

The model **arcee-ai/Llama-3.1-SuperNova-Lite** itself has its own special tokens. Its tokenizer's special tokens map shows:

- **BOS (beginning of text):** `"<|begin_of_text|>"` (ID 128000)
- **EOS (end of text):** `"<|eot_id|>"` (the Arcee model's end-of-text token) [5] [6].

In fact, the model's `tokenizer_config.json` lists `"<|eot_id|>"` as the `eos_token` [6]. (The config also shows the EOS token IDs include `128009` for `<|eot_id|>`, as well as other special tokens like `<|end_of_text|>` and `<|eom_id|>`, but the key point is that `<|eot_id|>` **is the model's default EOS marker**.)

Historical `eos_token` in Axolotl's ShareGPT

In older Axolotl versions (when `sharegpt` was supported), the framework did *not* automatically override the model's EOS. Instead, if you did **not** set any `special_tokens.eos_token` in your config, Axolotl would effectively use the tokenizer's default EOS. For Llama-based models this often meant `"</s>"`, and for GPT-style models it meant `"<|endoftext|>"`. (For example, one GitHub issue notes that when tuning GPT-like models via ShareGPT format, each conversation should end with `<|endoftext|>` [7] [8].) In the SE6446 example, no `eos_token` was specified, so the default EOS (`<|eot_id|>` for this model [6]) would have been used implicitly at the end of each formatted prompt. In other words, the old ShareGPT processing simply relied on the model's native EOS token to terminate the conversation.

Key point: *Earlier Axolotl (v0.4.x) did not insert* `<|im_end|>` *by default – it just ended sequences with whatever the tokenizer's eos_token was. In practice, that meant the Arcee model's default EOS (e.g.* `<|eot_id|>`*) marked the end of the conversation.*

ChatML Template and `<|im_end|>`

Axolotl's **ChatML** template (used here via `conversation: chatml`) formats each turn as `"<|im_start|>{role}\n{content}<|im_end|>\n"`. By convention, `<|im_end|>` denotes the end of each message or turn [8]. The Axolotl documentation explicitly notes that ChatML uses `<|im_end|>` to terminate turns, and **recommends** setting `special_tokens: { eos_token: <|im_end|> }` so the tokenizer treats `<|im_end|>` as the model's EOS in training [8]. Indeed, an example in the docs shows adding:

```
special_tokens:
  eos_token: <|im_end|>
```

to the config so that each `<|im_end|>` inserted by the template counts as an EOS when training [8]. Another Axolotl note emphasizes that *your* `tokenizer.eos_token` *should match the EOS/EOT token in the chat template* (otherwise you must set it manually) [4].

Thus, under the new `chat_template` approach, one typically sets `<|im_end|>` as the `eos_token` so that the model learns to predict an EOS at the end of each message (i.e. end of turn).

Comparing Old vs. New EOS Usage

- **Original (ShareGPT) behavior:** The conversation text was split into turns, but the only EOS was effectively the model's default EOS (e.g. `<|eot_id|>` for this Llama model). No `<|im_end|>` token was involved unless the dataset texts already contained it. Training would end a conversation when the default EOS token appeared.

- **New (ChatML) approach:** Every turn explicitly ends with `<|im_end|>` in the formatted prompt. To align this with training, we set `eos_token: <|im_end|>`. The tokenizer is then modified (under the hood) to recognize `<|im_end|>` (ID 128257 in examples) as an EOS marker [8]. This means the model is trained to consider `<|im_end|>` the end-of-sequence for each message.

Equivalence: `<|im_end|>` is *not* the same string as the model's original EOS (`<|eot_id|>` or `</s>`). However, when using ChatML it functions as an end-of-turn marker. If your goal is to replicate exactly the old setup (where only the model's native EOS ended the sequence), you could have instead set `eos_token: <|eot_id|>` (the model's default) or left it unset. But the new Axolotl docs advise using `<|im_end|>` to match the ChatML template [8] [4].

Recommendation

In summary:

- The old `sharegpt` config implied using the model's built-in EOS token (e.g. `<|eot_id|>` for Llama-3.1-SuperNova-Lite) at the end of each prompt.
- The modern `chat_template` setup expects `<|im_end|>` as the turn-ending token, and Axolotl now **recommends** explicitly setting it as the `eos_token` [8] [4].
- Keeping `"<|im_end|>"` as `eos_token` is therefore *safe and correct* for the ChatML template: it ensures the tokenizer treats the template's end-of-message delimiter as the EOS during training [8].

However, it is not exactly the same as the model's original EOS. If for some reason you wanted the model to use its native end-of-text token (for example, to exactly mimic prior behavior), you could set `eos_token` to the model's own token (`<|eot_id|>` [6] or `"<|end_of_text|>"`). But when using the ChatML template, using `<|im_end|>` is the more faithful choice to the prompt format (and is the approach documented by Axolotl [8] [4]).

Sources: Axolotl official docs and code (chat template guidelines) [8] [4]; model config for arcee-ai/Llama-3.1-SuperNova-Lite (shows `<|eot_id|>` as EOS) [6]; example config and dataset card from SE6446's model (v0.4.1 Axolotl) [1] [2].

---

[1] SE6446/Llama-3.1-SuperNova-Lite-Reflection-V1.0 · Hugging Face
https://huggingface.co/SE6446/Llama-3.1-SuperNova-Lite-Reflection-V1.0

[2] README.md · SE6446/MAGllama_Sharegpt at main
https://huggingface.co/datasets/SE6446/MAGllama_Sharegpt/blob/main/README.md

[3] dataset type sharegpt no longer works · Issue #1649 · axolotl-ai-cloud/axolotl · GitHub
https://github.com/axolotl-ai-cloud/axolotl/issues/1649

[4] Conversation – Axolotl
https://axolotl-ai-cloud.github.io/axolotl/docs/dataset-formats/conversation.html

[5] [6] special_tokens_map.json · arcee-ai/Llama-3.1-SuperNova-Lite at main
https://huggingface.co/arcee-ai/Llama-3.1-SuperNova-Lite/blob/main/special_tokens_map.json

[7] phi-2 training tokenizer bug when using sharegpt format · Issue #1357 · axolotl-ai-cloud/axolotl · GitHub
https://github.com/OpenAccess-AI-Collective/axolotl/issues/1357

8 Dataset Formats – Axolotl
https://axolotl-ai-cloud.github.io/axolotl/docs/dataset-formats/