	keep_history: Indicates whether to retain the historical context during inference. Set to 1 for multi-turn conversations.

Table 3-18 Explanation of RKLLMLoraParam Structure

Definition	RKLLMLoraParam
Introduction	Used to define the parameters for using LoRA during inference.
Struct Fields	***const char lora_adapter_name*:*** The name of the LoRA used during inference.

Table 3-19 Explanation of RKLLMPromptCacheParam Structure

Definition	RKLLMPromptCacheParam
Introduction	Used to define the parameters for using Prompt Cache during inference.
Struct Fields	***int save_prompt_cache:*** Indicates whether to save the Prompt Cache during inference. 1 means it is required, and 0 means it is not. ***const char prompt_cache_path*:*** Path to save the Prompt Cache. If not set, it defaults to “./prompt_cache.bin”.

Here is an example of using RKLLMPromptCacheParam for inference:

```
// 1. Initialize and set LoRA parameters (if needed)
RKLLMLoraParam lora_params;
// Specify the LoRA model name
lora_params.lora_adapter_name = "test";
// 2. Initialize and Set Prompt Cache Parameters(if needed)
RKLLMPromptCacheParam prompt_cache_params;
// Enable saving Prompt Cache
prompt_cache_params.save_prompt_cache = true;
// Specify cache file path
prompt_cache_params.prompt_cache_path = "./prompt_cache.bin";
rkllm_infer_params.mode = RKLLM_INFER_GENERATE;
rkllm_infer_params.lora_params = &lora_params;
rkllm_infer_params.prompt_cache_params = &prompt_cache_params;
```

3.2.4 Initialize Model

Before initializing the model, it is necessary to define the LLMHandle handle in advance. This handle is used for the initialization, inference, and resource release processes of the model. It's important to note that only by unifying the LLMHandle handle object across these three processes can the inference