RKLLM will detect the parts of the input that are the same as those in the prompt_cache from the beginning. If your input format is fixed as PROMPT_PREFIX + text + PROMPT_POSTFIX, you can generate the Prompt Cache for just the PROMPT_PREFIX part. After loading, you can reuse this part of the result in subsequent inferences.

RKLLM supports generating multiple Prompt Cache files. When different Prompt Cache files are needed, you can simply load the corresponding file. If you need to switch to another Prompt Cache file or no longer need the loaded Prompt Cache, please explicitly call the rkllm_release_prompt_cache interface to release it.

Here is an example Code for Loading Prompt Cache:

```
// Initialize and set the Prompt Cache parameters, then call the run
interface to generate the Prompt Cache file.

RKLLMPromptCacheParam prompt_cache_params;
// Whether to save the prompt cache
prompt_cache_params.save_prompt_cache = true;
// If you need to save the prompt cache, specify the absolute path of
the cache file.
prompt_cache_params.prompt_cache_path = "/data/prompt_cache.bin";
rkllm_infer_params.prompt_cache_params = &prompt_cache_params;

rkllm_infer_params.mode = RKLLM_INFER_GENERATE;
rkllm_input.input_type = RKLLM_INPUT_PROMPT;
rkllm_input.prompt_input = (char *)prompt.c_str();
rkllm_run(llmHandle, &rkllm_input, &rkllm_infer_params, NULL);

// Load the prompt cache file to reduce prefill time.
rkllm_load_prompt_cache(llmHandle, "./prompt_cache.bin");
if (ret != 0) {
    printf("\nload Prompt Cache failed\n");
}

rkllm_run(llmHandle, &rkllm_input, &rkllm_infer_params, NULL);
```

3.2.10 KV Cache Management

RKLLM supports manual clearing of the KV cache, which can be used for both single-turn and multi-turn dialogues. When invoking the cache clearing function, if keep_system_prompt is set to 1, the system prompt (if present) will be retained; otherwise, the entire cache will be cleared.

The function definition is as follows:

Table 3-28 Interface Specification for the rkllm_clear_kv_cache Function