



图 1.1

FANxy 言喵软件（以下简称“本软件”）简易运行流程如图 1.1 所示，程序层面运行流程如下：

- 1. 通过 AudioCaptureThread 或 MicrophoneAudioCaptureThread 捕获音频数据
  - 2. 根据翻译模式选择不同处理路径：传统模式（语音识别+文本翻译）或语音翻译模式（直接语音翻译）
  - 3. 调用相应的百度 API 处理音频或文本数据
  - 4. 将翻译结果显示在主窗口和悬浮窗口中
  - 5. 支持通过虚拟麦克风播放翻译后的语音
- 部分算法细节不予展示。

本软件组成结构如下：

#### ## 1. 核心功能组件

##### #### 主应用程序类

- TranslatorApp `real\_time\_translator.py`：软件的主窗口类，管理整个应用的生命周期，包括 UI 初始化、线程管理、主题应用等核心功能。

##### #### 音频捕获组件

- AudioCaptureThread `real\_time\_translator.py`：负责系统音频捕获、音量计算和音频数据处理，支持实时监控系统声音。

- MicrophoneAudioCaptureThread `real\_time\_translator.py`：专门负责麦克风设备的音频捕获，支持设备选择、音频录制和音量监控。

##### #### 翻译处理组件

- SpeechTranslateProcessThread `real\_time\_translator.py`：实现语音翻译的核心逻辑，包括音频队列管理、音频预处理、百度语音翻译 API 调用和虚拟麦克风播放等功能。

- SpeechRecognitionThread `real\_time\_translator.py`：负责语音识别功能，处理音频数据并调用百度语音识别 API 获取文本。

- TranslationThread `real\_time\_translator.py`：处理文本翻译请求，调用翻译 API 进行文本转换。

#### ## 2. 用户界面组件

##### #### 主窗口与控制面板

- 主窗口包含语言选择（源语言/目标语言）、启动/停止按钮、音频源选择等控制元素。

- 音量监控面板显示系统音量和麦克风音量实时状态。

- 文本显示区域用于展示原文和翻译结果。

##### #### 悬浮翻译窗口

- FloatingTranslationWindow `real\_time\_translator.py` : 提供两种显示模式——字幕模式 (宽屏单句显示) 和小框模式 (紧凑多句显示), 支持拖拽和双击隐藏。

### #### 对话框组件

- SettingsDialog `real\_time\_translator.py` : 设置对话框, 提供麦克风/扬声器选择、翻译模式、悬浮窗模式、音频格式和界面主题等配置选项。
- ApiKeyDialog `real\_time\_translator.py` : API 密钥配置对话框, 引导用户输入翻译和语音识别 API 的密钥信息。

### ## 3. 工具与辅助类

#### #### 配置与主题管理

- ConfigManager : 管理应用配置, 负责加载和保存用户设置。
- ThemeManager : 管理界面主题, 支持浅色主题、深色主题和自定义背景主题切换。

#### #### 音效管理

- SoundEffectManager `real\_time\_translator.py` : 管理界面交互音效, 支持指定扬声器设备、音量控制和并发播放。
- SoundButton `real\_time\_translator.py` : 集成音效功能的自定义按钮组件。

### ## 4. API 集成组件

软件集成了多个 API 服务:

- 翻译 API : 用于文本翻译功能
- 语音识别 API : 用于将语音转换为文本
- 语音翻译 API : 直接将一种语言的语音翻译为另一种语言的语音

### ## 5. 音频处理与设备管理

- 支持系统音频捕获和麦克风输入
- 虚拟音频设备管理 (支持 VB-Audio Virtual Cable)
- 音频预处理功能 (预加重、降噪、归一化)
- 音频格式转换 (PCM/WAV)
- 动态音量阈值调整算法

### ## 6. 部署与依赖管理

- get-pip.py `get-pip.py` : pip 25.1.1 版本的安装脚本, 作为备用工具确保依赖环境的完整性。
- requirements.txt `requirements.txt` : 定义项目依赖包, 包括 pyaudio、soundcard、requests、numpy、PyQt5 等。
- start.bat `start.bat` : 应用启动脚本, 负责激活虚拟环境、设置路径和启动主程序。