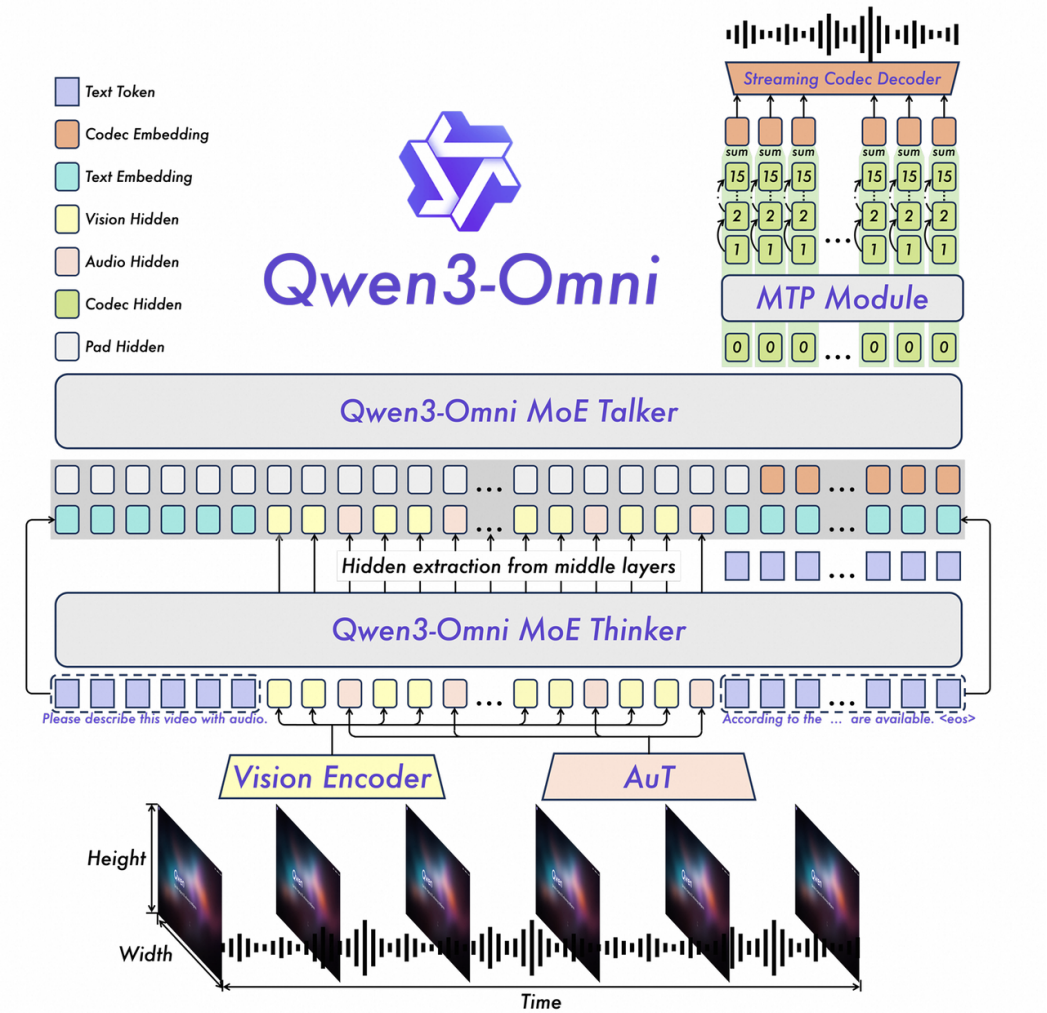
方案

瘦身（基于Qwen3-Omni-30B-A3B）



1. 视觉模态剥离

（1）**剥离视觉编码器**：在Qwen3-Omni架构中，直接移除视觉编码器及其对应的视觉Projector层，修改config中架构，在模型文件中移除对应参数。剥离这些模块后，还可能移除所有对应视觉Token的Embedding槽位。

（2）**重写forward**：重构模型的forward函数，删除处理视觉张量的所有逻辑分支。在处理多模态交错输入时，由于已经移除了视觉组件，需要逻辑上忽略视频帧占位符，仅保留音频和文本路径。

2. 基于MoE模型的音频无关专家剪裁

在稀疏MoE架构中，专家往往会在训练过程中产生领域特化，因此可以考虑裁剪掉那些专门负责处理图像/视觉信息的专家，降低thinker模型参数量。

参考REAP（https://www.alphaxiv.org/abs/2510.13999）

The above analysis demonstrates that the functional output space of a SMoE layer is defined by the *coordinated behaviour* of the router and experts. An expert's total contribution to its layer's output is determined by both its gate-value, $g_k(x)$, and the magnitude of its output vector, $\|f_k(x)\|_2$. However, naive frequency-based pruning fails to consider these properties. Intuitively, pruning experts which contribute minimally to the layer output minimizes the difference between the original and pruned layer outputs. Let $h(x)$ be the original output and $\bar{h}_{\setminus j}(x)$ be the output after pruning expert j and re-normalizing the remaining router weights. The error induced by pruning expert j is

$$\Delta\bar{h}_{\setminus j}(x) := h(x) - \bar{h}_{\setminus j}(x) = \sum_k g_k(x) f_k(x) - \sum_{k\neq j} \frac{g_k(x)}{1-g_j(x)} f_k(x). \tag{7}$$

Re-normalization of the router weights after pruning expert j modulates all other remaining expert outputs, making direct minimization of Δh_j complex. However, since our goal is to prune unimportant experts, we can reasonably assume their gate-values are small when active $\mathbb{E}_{x\sim\mathcal{X}}[g_j(x)] \ll 1$. Under this assumption, the weight re-normalization factor is negligible, i.e., $1-g_j(x) \approx 1$, and the error induced by pruning expert j is approximately equal to the expert's direct contribution to the layer output

$$\Delta\bar{h}_{\setminus j}(x) \approx \sum_k g_k(x) f_k(x) - \sum_{k\neq j} g_k(x) f_k(x)) = g_j(x) f_j(x). \tag{8}$$

To select which experts to prune, we propose a novel saliency criterion, REAP, which approximates an expert's importance by measuring its direct contribution to the layer's output magnitude. Specifically, the saliency score, S_j, is defined as the average of this contribution over tokens for which the expert is active

where S_j is the saliency of expert f_j and $\mathcal{X}_j$ is the set of inputs where $g_j(x) \in TopK(\mathbf{g}(\mathbf{x}))$.

$$S_j = \frac{1}{|\mathcal{X}_j|} \sum_{x\in\mathcal{X}_j} g_j(x) \cdot \|f_j(x)\|_2, \tag{9}$$

（1）专家激活路由分析：利用一个覆盖广泛音频场景（语音、音乐、环境音）的校准数据集，运行模型推理并记录每一层每个专家的路由权重。统计得到专家的平均路由权重分布。

（2）专家裁剪：如果某个专家在音频输入下的平均路由权重低于阈值 τ *（或者和REAP一样逐层删除一定比例的低激活专家）*，则可以判定该专家对音频模态是多余的，可将其进行移除。（可能需要训练：最小化剪裁前后的Token输出分布差异，或者在audio数据SFT重新训练router）

具体方法：基于三个集合进行剪枝，纯音频数据 (X_1)，纯视频数据 (X_2)，音频+视频混合数据 (X_3)，进行专家显著性分析。

目标：

1. **保留**：在纯音频数据(X_1) 上高激活的专家（纯Audio专家）。
2. **保留**：在混合数据(X_3)中负责音频处理的专家（Audio或Shared专家）。
3. **剪枝**：仅在纯视频数据 (X_2)上高激活，而在音频相关场景不活跃的专家（纯Video专家）。

专家显著性 (Expert Saliency)：定义单个专家 e 在某个数据集 D 上的基础显著性得分 $S(e, \mathcal{D})$ 。

- $g_e(x)$ 为路由网络分配给专家 e 的门控权重 。
- $\|h_e(x)\|_2$ 为专家 e 输出的激活值 L2 范数 (Activation Norm) 。

公式如下：

$$S(e, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x\in\mathcal{D}} (g_e(x) \cdot \|h_e(x)\|_2)$$

为了找到与 **Audio 最相关** 的专家，定义**音频亲和力得分 (Audio Affinity Score)**，记为 $\mathcal{A}_{audio}(e)$。定义 S_1, S_2, S_3 分别对应 X_1, X_2, X_3 上的专家显著性得分。

- 如果专家在 X_1 很活跃，必须保留。
- 如果专家在 X_3 很活跃，但同时在 X_2 也很活跃（且 S_1 很低），说明它在 X_3 中的活跃主要是因为视频部分，应被视为“视频专家”并剔除。
- 如果专家在 X_3 很活跃，但在 X_2 不活跃，说明它处理的是混合数据中的音频部分或跨模态对齐部分，应保留。

最终公式：

$$\mathcal{A}_{audio}(e) = S_1(e) + \lambda \cdot \text{ReLU}\left(S_3(e) - \beta \cdot S_2(e)\right)$$

参数解释：

- $S_1(e)$: 核心项。纯音频环境下的 REAP 显著性，直接决定该专家是否是“纯音频专家”。
- $S_3(e) - \beta \cdot S_2(e)$: 差分项。利用 X_2 作为“负样本”从 X_3 中剥离视频信号。
  - 如果 $S_3(e) \approx S_2(e)$，说明该专家在混合数据中的激活完全可以由视频数据解释，差分为 0，不增加音频权重。
  - 如果 $S_3(e) \gg S_2(e)$，说明该专家在混合数据中处理了非视频信号（即音频或交互），增加音频权重。
- $\text{ReLU}(\cdot)$: 确保非负。我们不希望视频专家的得分为负数从而扣除的分数，保持数值稳定性。
- λ : 混合数据的重要性权重（通常设为 1.0，如果更看重纯净音频能力可设为 0.5）。
- β : 视频去噪系数（通常设为 1.0，如果担心误删共享专家，可设为 0.8）。

计算出所有专家的 $\mathcal{A}_{audio}(e)$ 后，执行以下步骤：

1. **排序**：将所有专家按 $\mathcal{A}_{audio}(e)$ 从高到低排序。
2. **截断**：根据目标保留率（例如保留 50% 的专家），选择 Top-K 个专家。

$$\mathcal{E}_{keep} = \{e \mid \text{Rank}(\mathcal{A}_{audio}(e)) \leq K\}$$

3. **重组**：将保留的专家组成新的 Audio-MoE 模型，其余专家移除或置零。

3. 层剪枝

基于相邻层的hidden states相似度进行层剪枝

后训练/蒸馏

（可选）Stage1： 副语言信息增强

当前的音频模型往往面临“文本替代式推理”的问题，即模型倾向于根据语音转录的文本进行逻辑推演，而忽略了声音本身的副语言信息。参考Step-Audio-R1的MGRD（Modality-Grounded Reasoning Distillation）框架，我们可以实施以下方案，对qwen3-audio中的thinker模型进行微调：

方案一：短COT推理

1. **冷启动SFT**：收集包含副语言信息的音频样本，构建音频推理数据集，对模型进行SFT冷启。教师蒸馏对于每个音频样本，从多个教师模型采样，引导模型不仅给出答案，还要给出包含声学特征分析的推理过程。例如，“音频中的人语速较快、音调上扬，且伴随呼吸急促，这表明其处于焦虑状态”。然后结合llm-judge从教师模型生成的推理链中筛选出真正结合了副语言信息的输出。过滤准则包括：逻辑链条必须显式引用音质、音调轮廓或节奏等副语言信息。过滤掉仅使用文本语义的推理。
2. **on-policy RL**：通过RL，鼓励模型生成那些能够被声学事实验证的推理步骤，从而将“文本推理”转化为“原生音频思维”。llm-as-a-judge+rubrics reward：将captioner生成的音频描述作为参考，让一个文本模型作为judge，为thinker输出进行打分，衡量模型是否正确结合了不同副语言信息进行推理，每个任务可以为不同副语言信息分配不同权重。

方案二：非显示推理

同样是冷启动SFT+RL，不过不需要显式COT

1 . Audio Reward Model: 专门训练一个audio-to-text的奖励模型，用于判断模型输出的audio是否与用户输入audio的副语言信息匹配（比如是否考虑到了用户的情绪，音色等），用于数据合成/RL训练。

2 . Captioner+text-based RM：同时为user输入和assistant输出语音生成描述，用一个text-based LLM判断assistant输出是否与user匹配

Stage2：音频编码器与解码器适配

thinker更新后，需要同步提升音频编码器对副语言的理解和抗噪能力。

1. **合成副语言增强**：利用SynParaSpeech等框架，合成大量带有“笑声”、“叹息”、“语塞”等非言语符号的语音数据。通过这些合成数据，强制编码器学习如何从声学信号中提取细粒度的情绪和风格嵌入。

2. **噪声鲁棒性注入**：人工合成真实世界的干扰数据。利用房间脉冲响应（RIR）模拟物理空间的混响效果，并叠加各种背景噪声（如交通噪音、工厂噪音、现场人声噪音）。

   ◦ **构建方法**：采用SNR（信噪比）混合器，将干净语音与不同强度的环境噪音按随机比例混合，并应用随机裁剪和频率masking。

   ◦ **训练策略**：固定已经蒸馏好的LLM，仅训练音频编码器。通过最小化被干扰音频与原始干净音频在LLM表征层之间的KL散度，实现编码器的环境无关表征学习。