信也科技杯

学术指导单位
中国计算机学会自然语言处理专业委员会(CCF-NLP)

合作院校
复旦大学自然语言处理实验室(FudanNLP)

赛事合作
第15届CCF自然语言处理与中文计算国际会议(NLPCC2026)

第十一届信也科技杯
全球AI算法大赛

LISTEN BEYOND 循声知意

2026 FINVOLUTION GLOBAL DATA SCIENCE COMPETITION

TURN-TAKING MODELING IN CONVERSATION CHALLENGE

CONTENT

01 团队介绍

02 赛题分析

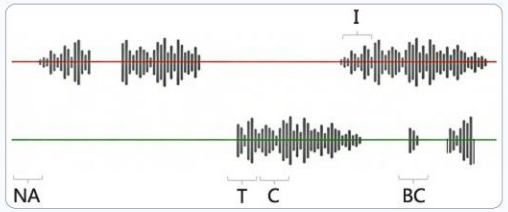
03 整体方案

04 总结

赛题分析

全双工语音交互时代，精准判断「何时开口/闭嘴」仍是核心瓶颈。本届信也科技杯全球 AI 算法大赛聚焦**汉语多方言对话的轮次交互建模**，要求基于双方对话历史音频预测未来窗口内的语音事件，在较短时延下判断用户意图，避免 AI 咄咄逼人或冷场，通过探索多模态信号赋予 AI 得体的「社交直觉」，最终实现自然流畅的人机交互。

下图展示了用户A（说话人，上面红线）和用户B（倾听者，下面绿线）正在进行的对话：



基于上述对话形式，我们将对话音频按 80ms 为单位切分为连续的时间片段（chunk）。每个 chunk 被标注为以下 5 种轮换事件之一：

标签	英文	含义	示例场景
C	Continuation	当前说话人继续说话	一方正在陈述观点
T	Turn Change	话轮切换，另一方接话	“你觉得呢？”→ 对方开始说话
BC	Backchannel	短回应，不夺取话轮	对方在说话时插入“嗯”、“对”等
I	Interruption	双方同时说话，争夺话轮	一方打断另一方
NA	Silence	双方均未说话	话轮间的短暂沉默

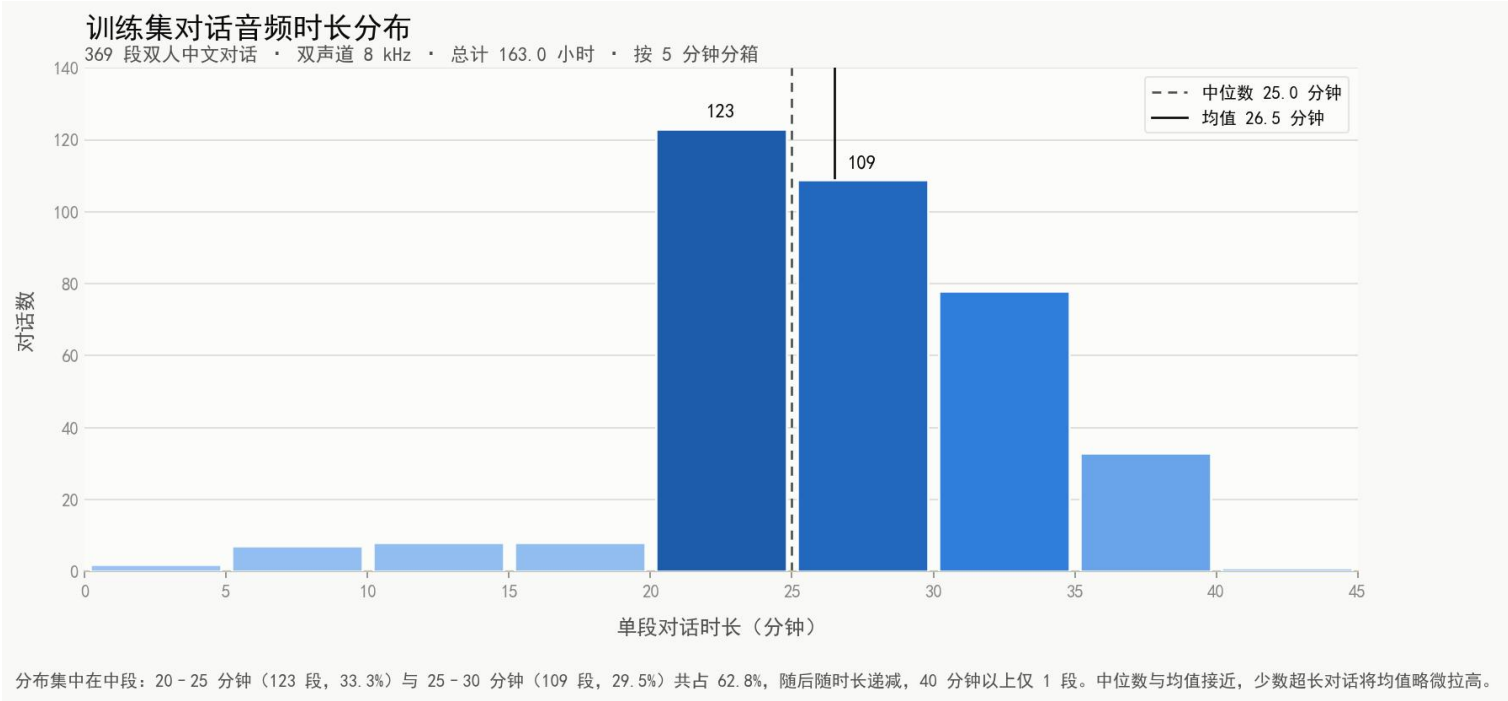
给定过去 30 秒的双声道对话音频作为上下文，模型需要预测紧随其后的 25个 chunk（共 2 秒）中是否出现相关语音事件。

数据分析

输入数据分为三种模态：分别为**text、context** 和 **audio。**

输出预测未来2s内是否出现 **C、T、BC、I、NA** 五种对话状态。

	对话总数	**对话时长**
训练集	369	右图分析
测试集A榜	1000	30s
测试集B榜	n	0-30s



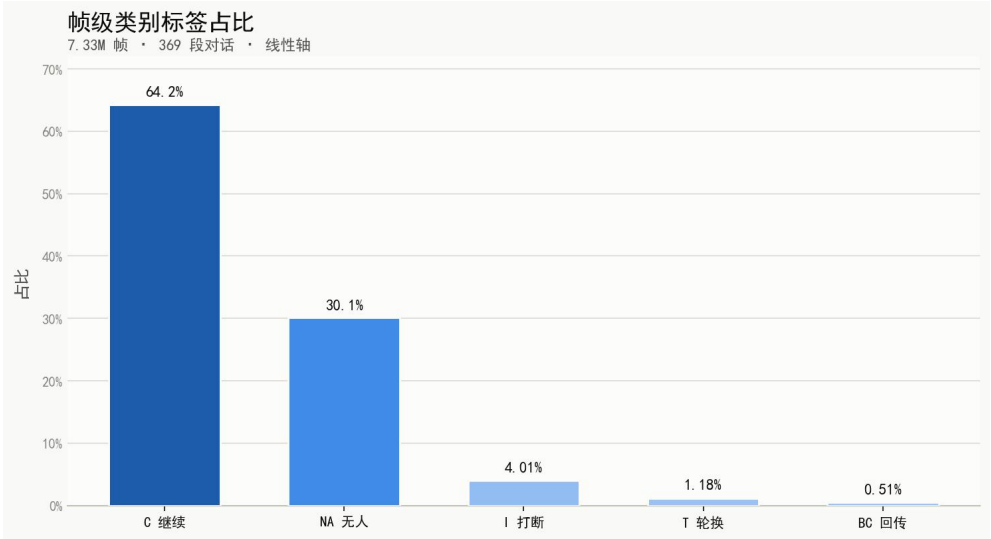
- **评测指标**
  - **F1 = 2*P*R/（P+R）**

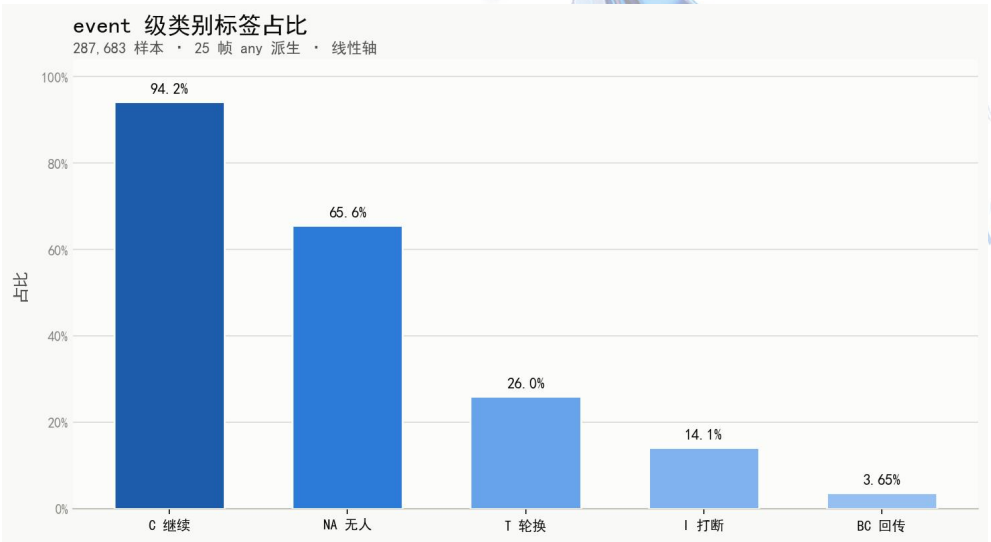
数据难点分析

- Audio 样本中大部分对话为方言
  - 如何引导模型在听不懂的状态下理解上下文

A：喂
B：喂，哎，龙哥。
A：哎，古天乐，听得到吧?
B：听见了，明天地暖开开呢。

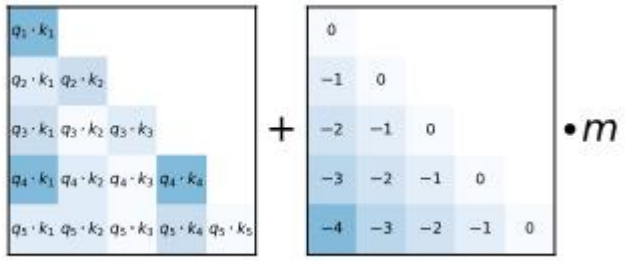
- 类别分布不均衡
  - Frame 级别和 Event 级别的类别标签分布不均衡
  - 在 stride=25 下统计标签占比



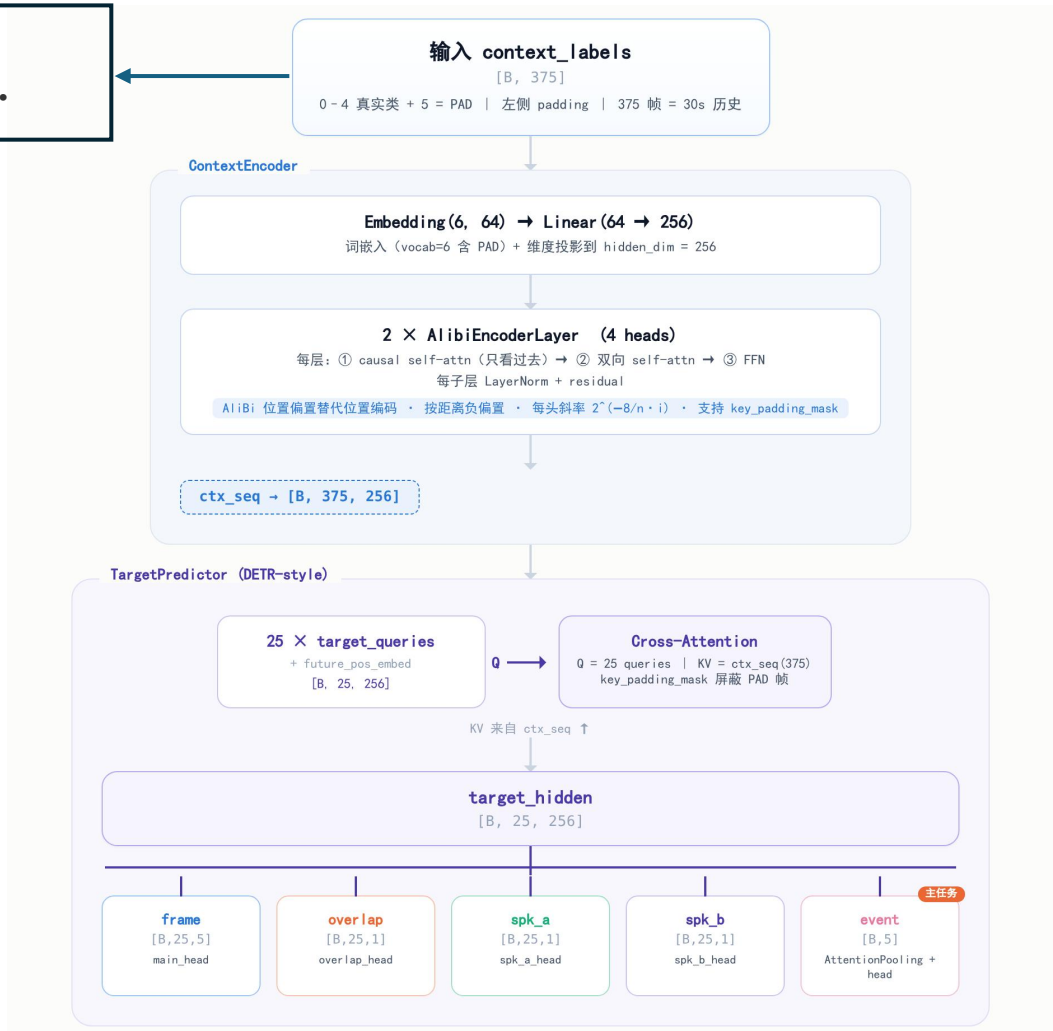


整体方案-Context only

- **AlibiAttention**
  - 任务本质是"**看历史 → 预测未来**",强烈依赖时序距离信息
  - 复赛的输入是不定时长的，训练时学到的"距离→注意力衰减"规律，对任意长度都成立
  - AlibiAttention的"随距离衰减"恰好匹配这个先验——**越近的帧天然获得更高注意力**,这个归纳偏置和任务吻合。



…55550124312…



整体方案-audio only

- **选用 Whisper-large-v3 模型**
  - 预训练数据规模大，鲁棒性足够强
    - VAP 本质上是 语义+韵律+交互 的任务
    - Whisper 的训练目标是 predict text，Encoder 学的是通用声学表示，不绑定 ASR 任务
    - 训练语料包含：普通话、粤语、台湾国语、闽南语、上海话、客家话等等
- **TransformStereo**
  - 特点：两路共享权重先各自做因果自注意力，再用 3 层对称交叉注意力做深度双向信息交换，最后用通道专属线性合并

整体方案-Audio only

- **模型输出 5 个头，但任务层级不同，分为 event 与 frame 两个 level:**

head	output_dim	label_source	loss_type	weight	role	core_function
event	[B, 5]	25帧 any() 派生	Weighted BCE	1.0	主任务	比赛目标:未来2s事件集合
frame	[B, 25, 5]	target_labels 原值	Focal CE(γ=2) + class_weight	0.2	辅助	强制逐帧分辨率
overlap	[B, 25, 1]	label ∈ {BC,I} 派生	BCE(pos_weight=20)	0.3	辅助	集中学"重叠"信号
spk_a	[B, 25, 1]	VAD 生成	BCE(pos_weight=5)	0.2	辅助	注入说话人A活跃状态
spk_b	[B, 25, 1]	VAD 生成	BCE(pos_weight=5)	0.2	辅助	注入说话人B活跃状态

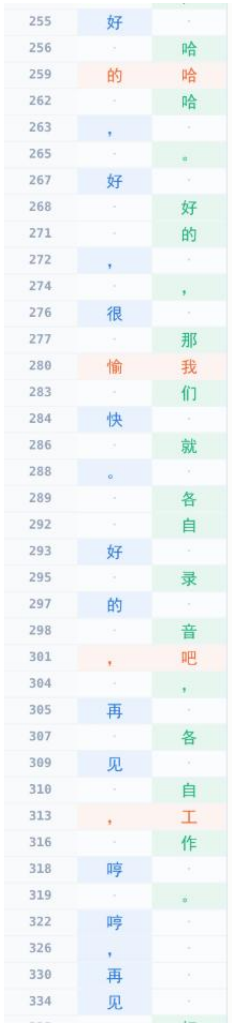
- **类别不均衡**
  - Event head 上面增加 label_weight，乘到 loss 上，通过梯度影响参数的学习

类	正样本率 p	1/p（理论逆频率）	cfg
C	94.17%	1.06	1.00
T	25.98%	3.85	2.80
BC	3.65%	27.38	26.40
I	14.15%	7.07	6.10
NA	65.55%	1.53	1.00

整体方案-text only

- **双流输入对齐**
  - 保留"谁在说"的**身份**信息
  - **重叠帧**不被压缩
  - 天然时间对齐，保留了**时间**信息
  - 交叉注意了能学到 **双声道交互** 模式
  - 对短促信号 BC 更为友好

- **选取 Qwen3-0.6B 模型**
  - 文本对 event 预测的收益小于音频
  - 不应该使用参数量过大模型，及易造成过拟合
  - 冻结骨干网络获取 hidden_state，不需要 lm_head 做生成，语义信息已经足够

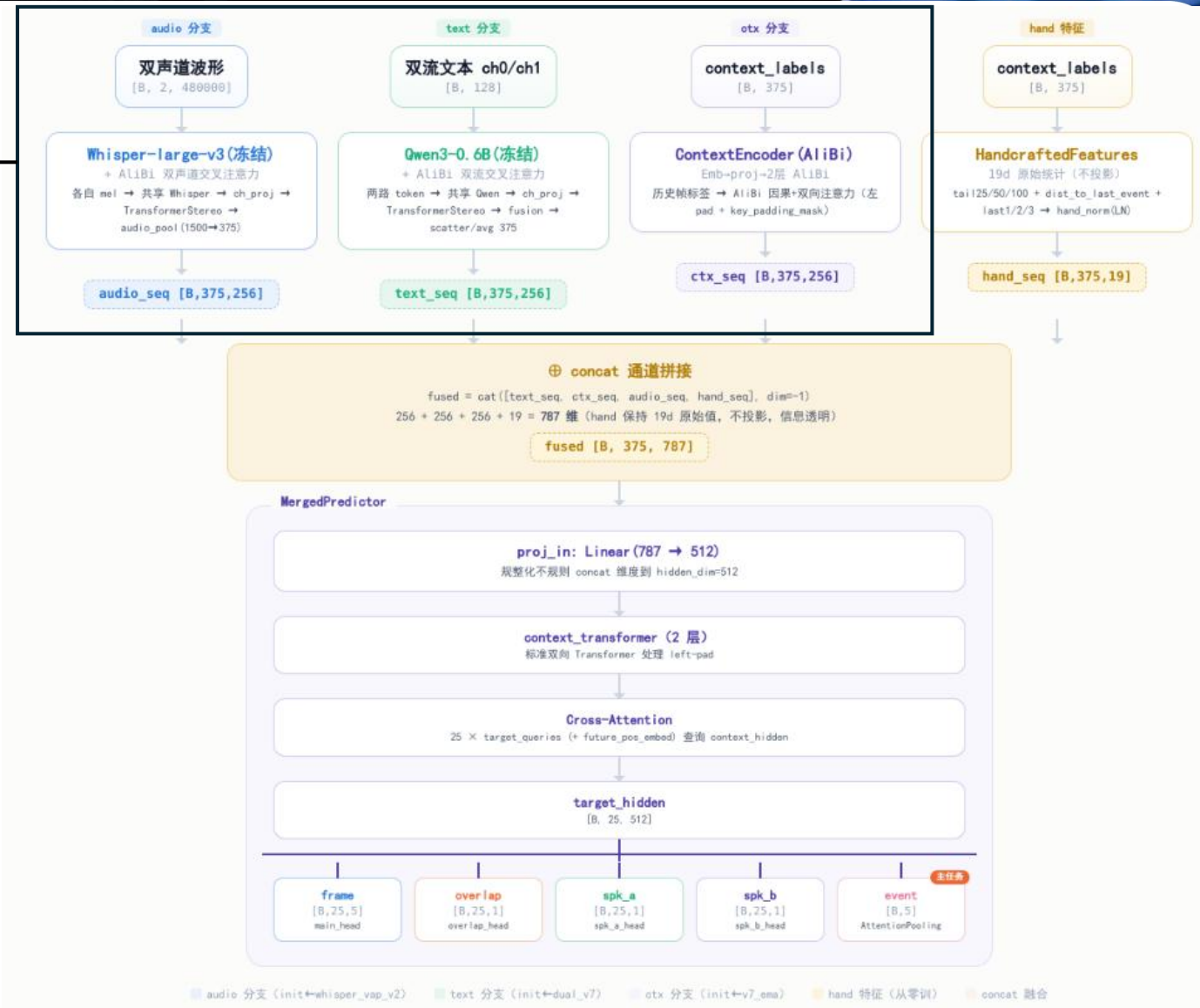


整体方案

	线下得分auc	线下得分f1	线上得分f1
Audio-only	**0.5925**	**0.8084**	**0.656**
Text-only	0.5789	0.8047	0.643
Context-only	0.5736	0.8091	0.646

整体方案-Merge_concat

- **Stage1**
  - 载入三种模态的 Best ckpt 并冻结
  - 只训练 Predictor 的参数
- **Stage2**
  - 解冻编码器用小学习率微调
  - 训练 Predictor 的参数
- **Merge concat 模式**
  - 不改变各分支内部结构，只在末端把三路特征拼起来。
  - 所以三个单模态的能力都被保留，让 MergedPredictor 学最优组合。

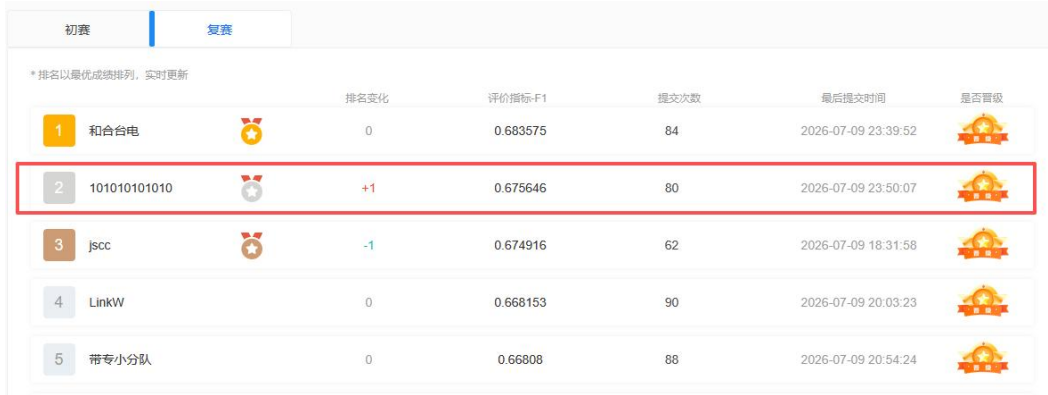


整体方案-Merge_hierarchical

- **Hierarchican Fusion**
- **Stage1：T 用 C 消歧**
  - T1 = T + CausalCrossAttn(Q=T, K=V=C)
  - text 做 query，ctx 做 key/value。让文本从上下文标签里查时序边界信息
- **Stage2：A 用 T1 消歧**
  - A1 = A + CausalCrossAttn(Q=A, K=V=T1)
  - audio 做 query，T1（已带 ctx 信息的 text） 做 K/V。这一步是 A 查已经消过歧的 T1。
- **Stage3：softmax 3-way 门控**
  - **逐帧、逐样本**：每个 80ms 帧、每个样本都有独立的 [wA, wT, wC]，不是全局固定比例。意味着模型能学到"这帧该信音频、那帧该信上下文"。
  - **softmax 竞争**：三权重和为 1，是竞争关系而非独立。某帧音频强则 wA↑、wT/wC↓，自动分配模态可信度。
  - **A1 是主干**，T1/C 是**门控残差**：即使门控把 wT/wC 学成 0，至少还有 A1，保证主干不被压垮。

整体方案-消融实验

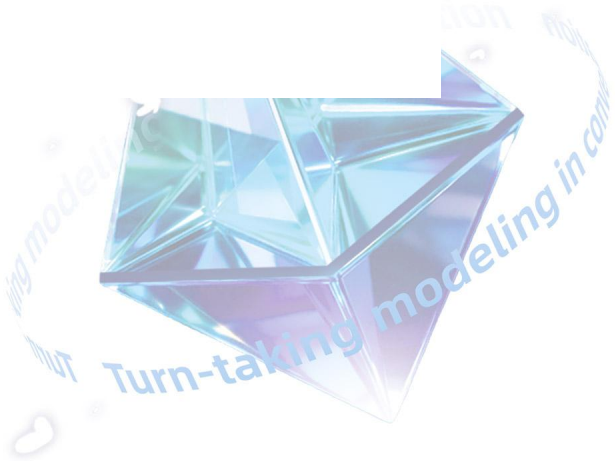
		线下得分auc	线下得分f1	opt_f1	线上得分f1
单模态	Audio-only	0.8084	0.5925	x	0.656
	Text-only	0.8047	0.5789	x	0.643
	Context-only	0.8091	0.5736	x	0.646
多模态	Merge-T-C	0.8291	0.6076	x	0.660
	Merge-concat-3	0.838	0.6192	0.669	0.665
	Merge-hiearc-3	0.8410	0.627	0.678	0.670
融合	双模型融合BC	x	x	x	0.6756



- 最终 **单模型** 得分排名 **第三**，融合两个模型排名 **第二**
- 模型参数大概在 1.24B，远小于 8B
- 未充分尝试融合策略
  - 单类别 BC 并集融合带来较高收益
- 未充分尝试线上的阈值优化

一些思考

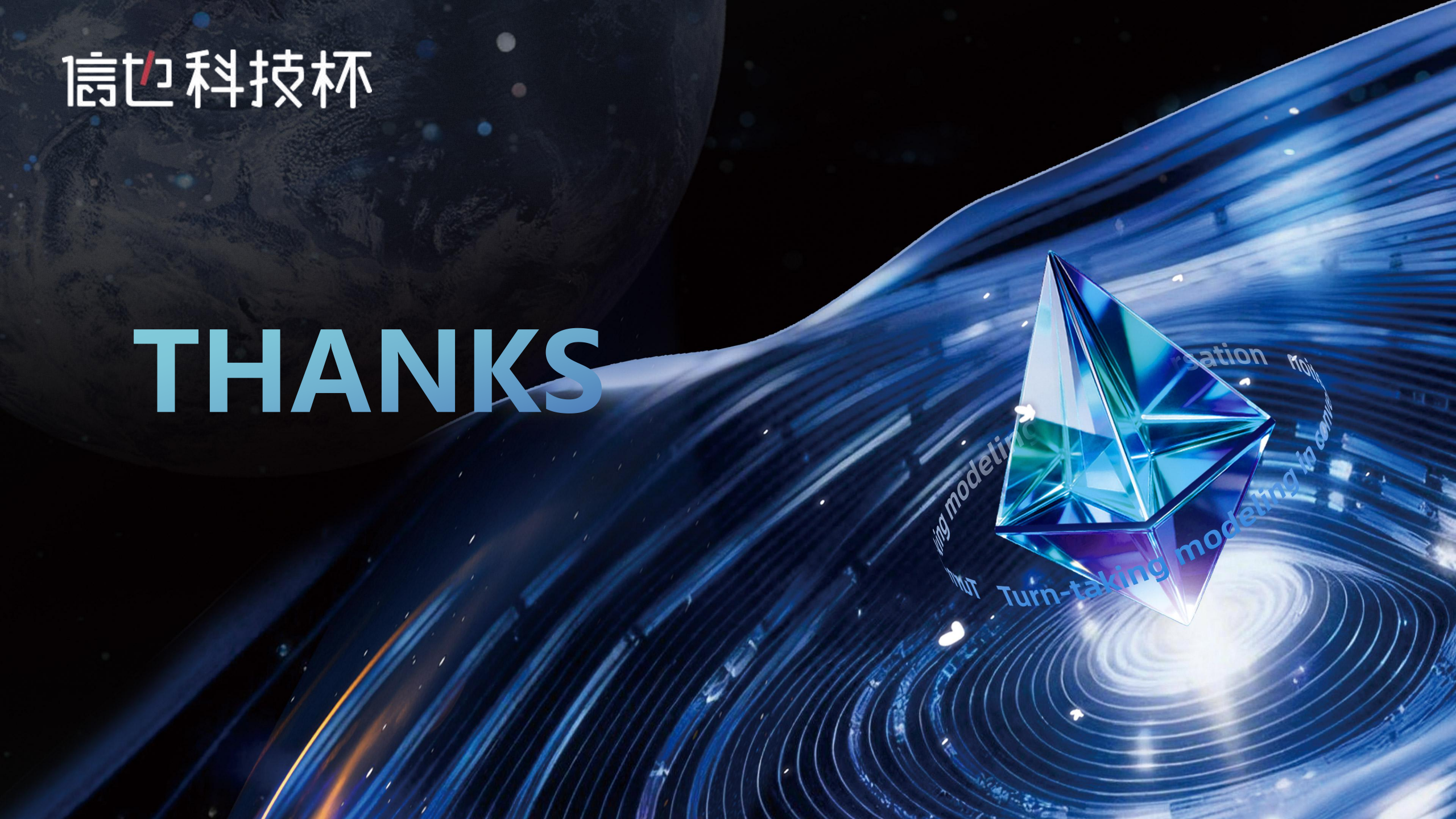
- 验证集中 BC 和 I 的 recall 偏低
  - 存在大量 30s 内无 BC 和 I 但是 event label 中有的样本
- C 和 NA 相对于另外三类样本数极多
  - 导致漏判和误判的样本失分相对极低（C提交全是1也接近满分）
  - 是否有更科学的指标，F1过于宽容
- 可以增加信息熵更大的模态，比如脸部表情，可以更精准的预测 2s 中 user 的状态
- 量化模型 attention 中 QKV bit，多轮对话使用 prefix caching 提升推理速度



Refence

- **Softmax 3-way 门控公式**

```
gate_in = torch.cat([A1, T1, ctx_seq], dim=-1)   # [B, 375, 3D=768]
gate_logits = self.gate_net(gate_in)             # [B, 375, 3]
wA, wT, wC = softmax(gate_logits, dim=-1)        # 三个权重, 和=1
fuse = A1 + wT·text_proj(T1) + wC·ctx_proj(ctx_seq)   # A1 主干 + 门控残差
```

信也科技杯

THANKS