

# 情感语音生成实验报告

XXX

XXXXXXXXXXXX

## 1 研究动机

本研究旨在针对现有的情感语音生成模型的不足，提出数据端、训练端以及推理端三个方面的改进。

### 1.1 数据端

情感语音合成在数据层面通常面临两类矛盾：一方面，带有可靠情绪标注或细粒度情感描述的语音数据规模有限；另一方面，情感可控 TTS 训练又要求模型同时学习“内容一致性”和“情绪表达差异”，尤其当情绪通过自然语言提示进行控制时，训练数据既要覆盖多样表达，又要保证读稿不走样。

为此，本实验在数据端引入一种“生成-校验-筛选”的自动化构建策略：先生成适合承载多种情绪的参考文本，并分别以不同情绪合成语音；再通过 ASR 将合成语音转写回文本，与参考文本计算词错误率（WER）作为一致性指标；只有当一致性满足阈值要求时才将样本加入训练集，否则丢弃或重试。该策略的核心目的在于：在扩大情感数据规模的同时，用可量化的文本一致性约束抑制“漏词、添词、改写”等现象，使模型获得更干净的“同文本多情绪”监督信号，从而更聚焦于情绪差异的学习而非内容噪声。

### 1.2 训练端

情感语音合成的第二阶段微调通常以自回归 next-token prediction 的交叉熵为核心目标：模型在给定包含情感描述与待合成文本的提示词后，学习生成与目标语音对应的声学 token 序列。该目标能够有效保证“内容可懂度”和“局部声学一致性”，但对“情感描述是否被模型稳定、可控地落实到语音表达”缺少显式约束，即在交叉熵主导的优化下，模型往往优先满足更容易被预测的内容与声学细节，而将情绪指令当作弱条件，从而出现情感可控性不足、同一情感提示下表达不稳定等问题。

为增强情感控制的约束力，本实验在训练过程的第二阶段情感微调中引入额外的“情感对齐”目标：将情感描述文本的表示与目标语音的情绪表示对齐，并与原有交叉熵联合优化。直观而言，该目标迫使模型在隐空间中把“情感描述”映射到与真实语音情绪一致的位置，从而提升情绪指令对生成结果的可控性与一致性。

### 1.3 推理端

仅依赖训练阶段的交叉熵与情感对齐约束，模型在可控性上仍存在两个常见瓶颈：其一，自然语言情感提示往往是细粒度、连续变化的，例如“克制但明显的兴奋”，而模型更容易学到粗粒度的情绪类别，导致同一类提示下强度与风格不稳定；其二，不同说话人、不同文本与不同韵律条件下，情绪表达的可迁移性较弱，推理时若希望对情绪进行增强、削弱、插值甚至擦除，通常需要重新微调或额外条件网络，成本较高且不灵活。

为此，本实验在推理端引入一种无需训练，仅在推理端可以即时插入的控制机制：激活引导。其核心思想是：在不改动主模型参数的前提下，通过对语音生成模型的解码器内部关键层隐状态施加可控扰动，使生成语音的情绪朝指定方向偏移，从而实现连续、可解释的强度调节。同时，为了把“细粒度情绪描述文本”稳定映射到可用的控制量，本实验额外引入情感量化器：将输入的情绪描述量化为情绪类型（如 happy、sad 等）与强度（0 ~ 1），从而在推理时以统一接口驱动激活注入，实现“文本提示 → 强度可控的情绪操控”。

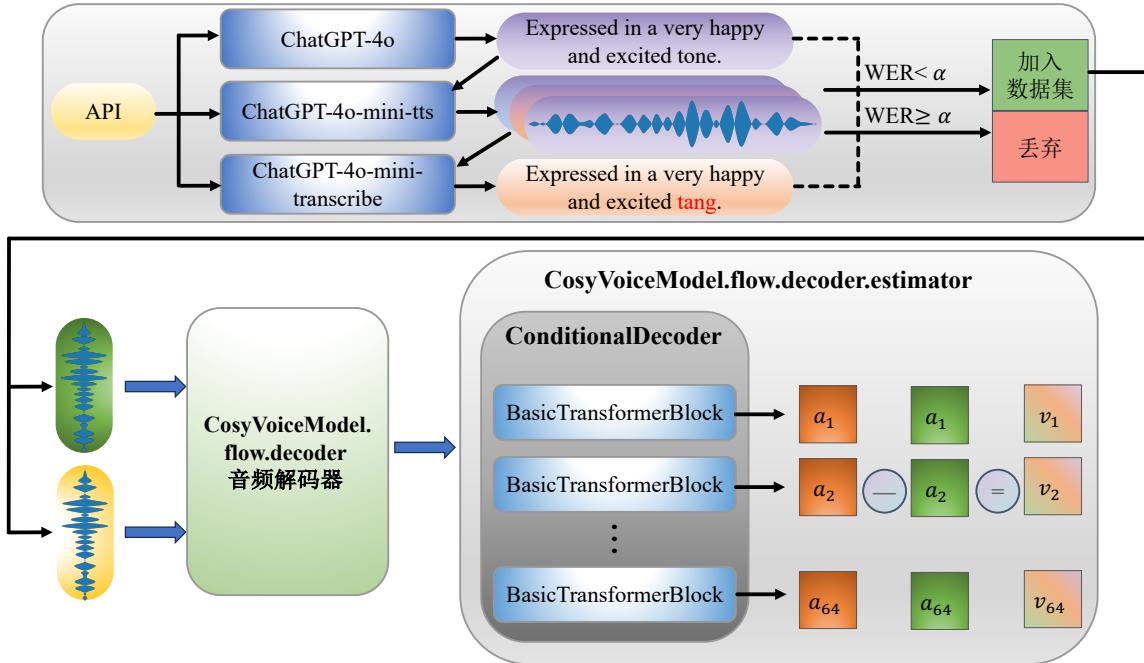


图 1: 数据集自动化构建框架 (上) 与情感激活提取架构图 (下)

## 2 方法

接下来将分别介绍数据端-“通过调用 API 构造高质量情感音频数据，为情感特征提取做准备”、训练端-“对音频的情感特征与情感描述文本特征加上一个显示约束，提高情感可控性与情感一致性”、推理端-“从 CosyVoice 解码器的输入残差流激活中统计‘原情绪 – 目标情绪’的差分均值方向，并在推理时用 hook 将该方向按可调强度注入相同层”的具体方法描述。

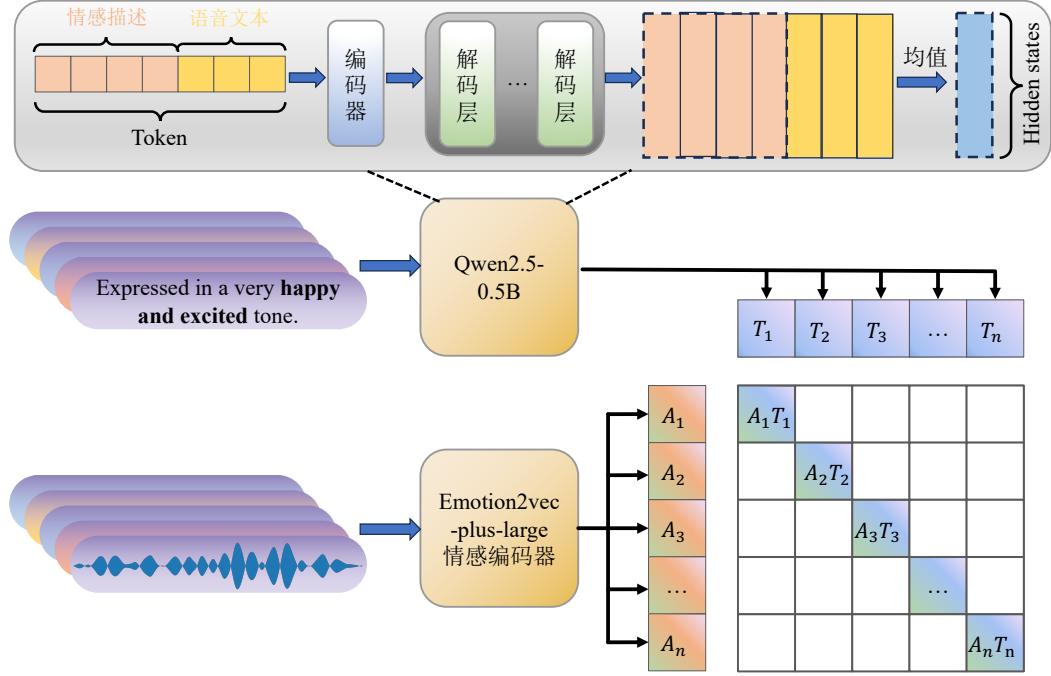


图 2: 语音情感特征-细粒度情感描述对比学习架构图

## 2.1 数据端

数据构建流程如图 1 上半部分所示，可概括为以下步骤：

- 参考文本生成：** 使用大语言模型生成一条“情绪可塑性”较强的英文句子作为参考文本，长度与标点受控，以便在 neutral、happy、sad 等不同情绪下均能自然朗读，并同步生成每种情绪对应的韵律与情感控制描述，仅约束语音学层面的语速、音高范围、能量与停连等，不引入额外语义内容。
- 多情绪语音合成：** 对同一参考文本，采用 TTS 模型分别合成 neutral、happy、sad 等情绪的语音，可在不同 voice 上生成以增加说话风格多样性。合成时将“情绪描述 + 强度”等控制信息注入指令，要求逐字朗读参考文本，不允许增删改写。
- ASR 转写校验：** 使用 ASR 模型对每条合成语音进行转写，得到假设文本。
- WER 过滤：** 计算参考文本与转写文本之间的 WER，并依据阈值筛选样本。WER 基于词级 Levenshtein 对齐，定义为

$$WER = \frac{S + D + I}{N}, \quad (1)$$

其中  $S$  为替换词数， $D$  为删除词数， $I$  为插入词数， $N$  为参考文本的词数。仅当  $WER < \alpha$  时保留样本，在此过程对 neutral 设置更严格阈值，对情绪更强的 happy/sad 适度放宽阈值；否则丢弃或触发重试合成。

经过上述过滤后，最终得到“同一文本在不同情绪下的多条语音”样本（例如 neutral-happy-sad 三元组），其优点在于：文本内容一致性较强，情绪差异主要由韵律与音色变化体现，能够为后续情感可控 TTS 提供更稳定的监督信号。

## 2.2 训练端

训练端的对比学习架构如图 2 所示。设一个 batch 大小为  $B$ 。对于第  $i$  条样本，记文本侧情感向量为  $\mathbf{t}_i \in \mathbb{R}^D$ ，音频侧情绪向量为  $\mathbf{a}_i \in \mathbb{R}^D$ 。

**(1) 音频侧情绪表示** 对每条训练样本的目标语音，使用冻结的语音情绪表征模型 emotion2vec-plus-large 提取 utterance-level embedding，得到  $\mathbf{a}_i$ 。为降低训练时的计算开销， $\mathbf{a}_i$  可离线预计并缓存，训练阶段按样本索引查表读取。

**(2) 文本侧情感表示** 将输入提示中的“情感描述”对应 token 区间记为  $[s_i, e_i]$ 。令大语言模型最后一层 hidden states 为  $\mathbf{H} \in \mathbb{R}^{B \times T \times H}$ ，则对区间做均值池化：

$$\mathbf{h}_i = \frac{1}{e_i - s_i + 1} \sum_{t=s_i}^{e_i} \mathbf{H}_{i,t} \in \mathbb{R}^H. \quad (2)$$

随后通过可学习线性投影  $W \in \mathbb{R}^{D \times H}$  得到

$$\mathbf{t}_i = W\mathbf{h}_i \in \mathbb{R}^D. \quad (3)$$

并对  $\mathbf{t}_i, \mathbf{a}_i$  做  $L_2$  归一化，使后续点积相似度等价于余弦相似度。

**(3) 相似度矩阵与 logits** 对 batch 内所有文本与音频 embedding 构造相似度矩阵：

$$s_{ij} = \mathbf{t}_i^\top \mathbf{a}_j. \quad (4)$$

采用可学习的 logit scale 替代固定温度系数：

$$\alpha = \exp(\gamma), \quad \gamma \in \mathbb{R}. \quad (5)$$

为避免数值不稳定，对  $\alpha$  做上界裁剪：

$$\alpha \leftarrow \min(\exp(\gamma), \alpha_{\max}), \quad \alpha_{\max} = 100. \quad (6)$$

对应 text→audio 的 logits 为

$$\ell_{ij}^{(t)} = \alpha s_{ij}. \quad (7)$$

(4) InfoNCE 对齐损失 将正确匹配对  $(i, i)$  作为正样本，其余  $(i, j), j \neq i$  作为负样本，定义 text→audio 的对比损失：

$$\mathcal{L}_{t \rightarrow a} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\ell_{ii}^{(t)})}{\sum_{j=1}^B \exp(\ell_{ij}^{(t)})}. \quad (8)$$

同理可得 audio→text 的对比损失  $\mathcal{L}_{a \rightarrow t}$ ，最终对齐损失取二者均值：

$$\mathcal{L}_{align} = \frac{1}{2} (\mathcal{L}_{t \rightarrow a} + \mathcal{L}_{a \rightarrow t}). \quad (9)$$

该形式与对比学习中常用的 InfoNCE 目标一致。

(5) 联合训练目标 第二阶段训练的最终目标为交叉熵损失与对齐损失的线性组合：

$$\mathcal{L}(\theta) = \mathcal{L}_{CE}(\theta) + \lambda \mathcal{L}_{align}(\theta), \quad (10)$$

其中  $\lambda$  为超参数，用于平衡内容建模与情感对齐约束。通过引入  $\mathcal{L}_{align}$ ，模型在保持原有自回归生成能力的同时，获得了显式的“情感描述 → 语音情绪表达”对齐监督，从而提升情感可控性与一致性。

## 2.3 推理端

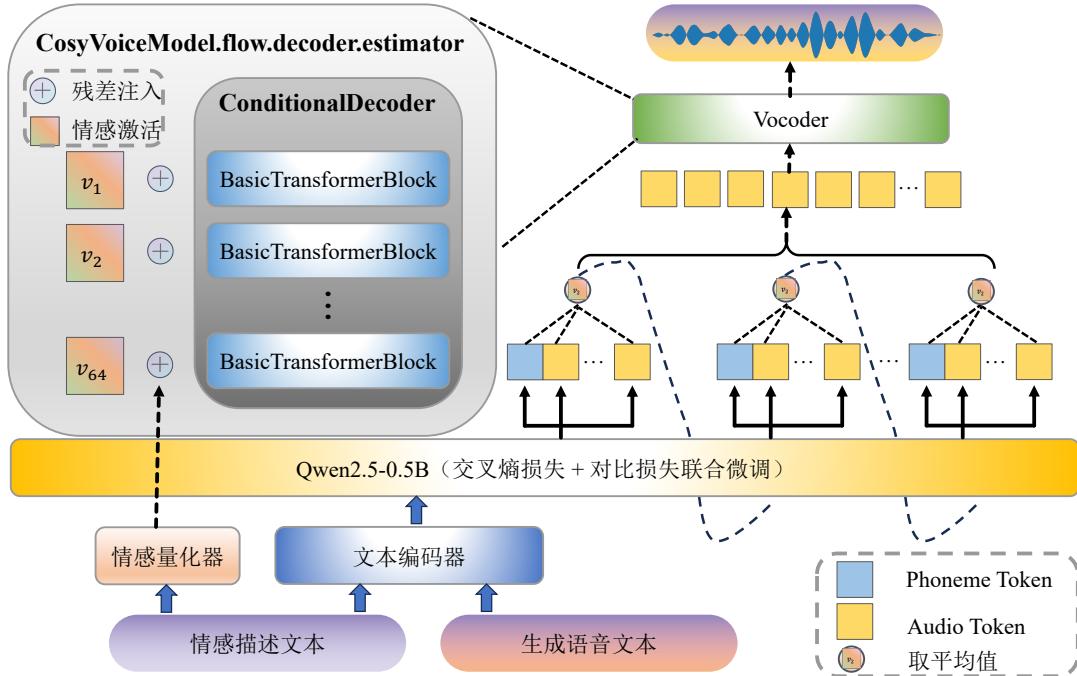


图 3: 推理时注入情感激活值架构图

推理端方法分为两部分：第一，离线提取情绪激活 steering 方向，如图 2 下半部分所示；第二，在线推理时的激活注入与强度控制，如图 3 所示。

(1) 离线提取情绪 steering 方向 设语音生成模型的解码器的 estimator 中共有  $L$  个 Transformer block (本实验使用的为 CosyVoice 模型, 用于解码 Tokens 生成语音,  $L = 64$ )。在第  $l$  个 block 的输入处采样其隐状态, 记为

$$\mathbf{h}^{(l)} \in \mathbb{R}^{B \times T \times C}, \quad (11)$$

其中  $B$  为 batch size,  $T$  为序列长度,  $C$  为 hidden size。为消除  $B, T$  可变带来的尺度差异, 对其做均值池化得到层向量:

$$\mathbf{a}^{(l)} = \text{MeanPool}(\mathbf{h}^{(l)}) = \frac{1}{BT} \sum_{b=1}^B \sum_{t=1}^T \mathbf{h}_{b,t,:}^{(l)} \in \mathbb{R}^C. \quad (12)$$

令中性参考集合为  $\mathcal{D}_{neu}$ , 目标情绪  $e$  的参考集合为  $\mathcal{D}_e$ , 分别计算每层均值:

$$\bar{\mathbf{a}}_{neu}^{(l)} = \frac{1}{|\mathcal{D}_{neu}|} \sum_{i \in \mathcal{D}_{neu}} \mathbf{a}_i^{(l)}, \quad \bar{\mathbf{a}}_e^{(l)} = \frac{1}{|\mathcal{D}_e|} \sum_{j \in \mathcal{D}_e} \mathbf{a}_j^{(l)}. \quad (13)$$

用“均值差”定义第  $l$  层的情绪 steering 向量:

$$\mathbf{u}_e^{(l)} = \bar{\mathbf{a}}_e^{(l)} - \bar{\mathbf{a}}_{neu}^{(l)} \in \mathbb{R}^C, \quad (14)$$

并进行逐层  $L_2$  归一化得到单位方向:

$$\mathbf{v}_e^{(l)} = \frac{\mathbf{u}_e^{(l)}}{\left\| \mathbf{u}_e^{(l)} \right\|_2 + \epsilon}. \quad (15)$$

将所有层堆叠可得该情绪的 steering 方向矩阵

$$\mathbf{V}_e = \left[ (\mathbf{v}_e^{(1)})^\top; \dots; (\mathbf{v}_e^{(L)})^\top \right] \in \mathbb{R}^{L \times C}, \quad (16)$$

并离线保存, 供推理阶段快速加载与调用。

(2) 情感量化器: 从描述到“类型 + 强度” 对输入的细粒度情绪描述文本  $d$ , 情感量化器输出情绪类型与强度:

$$e^* = \arg \max_e p(e | d), \quad \alpha = \sigma(g(d)) \in [0, 1], \quad (17)$$

其中  $p(e | d)$  为分类分布,  $g(d)$  为回归头输出,  $\sigma(\cdot)$  将强度压到  $[0, 1]$ 。最终在推理时选择  $\mathbf{V}_{e^*}$  作为 steering 方向, 并用  $\alpha$  控制注入幅度。

(3) 推理阶段: 激活注入与幅度稳定化 推理时在 estimator 的第  $l$  个 block 输入处, 将  $\mathbf{v}_{e^*}^{(l)}$  broadcast 到  $\mathbb{R}^{B \times T \times C}$ , 记为  $\hat{\mathbf{v}}_{e^*}^{(l)}$ 。引入情绪增强系数  $\alpha$  与情绪擦除系数  $\beta$ , 得到可控注入:

$$\tilde{\mathbf{h}}^{(l)} = \mathbf{h}^{(l)} + \alpha \cdot s \cdot \hat{\mathbf{v}}_{e^*}^{(l)}. \quad (18)$$

为抑制与移除情绪分量，可对 steering 方向上的投影进行削弱，改做法更稳定，避免直接相减导致幅度漂移：

$$\tilde{\mathbf{h}}^{(l)} \leftarrow \tilde{\mathbf{h}}^{(l)} - \beta \cdot \text{Proj}_{\hat{\mathbf{v}}_{e^*}^{(l)}}(\tilde{\mathbf{h}}^{(l)}), \quad \text{Proj}_{\mathbf{v}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{v} \rangle \mathbf{v}. \quad (19)$$

最后进行幅度稳定化：保留原始 token 范数  $n^{(l)} = \|\mathbf{h}^{(l)}\|_2$ ，对注入后的表示重标定：

$$\tilde{\mathbf{h}}^{(l)} \leftarrow \frac{\tilde{\mathbf{h}}^{(l)}}{\|\tilde{\mathbf{h}}^{(l)}\|_2 + \epsilon} \odot n^{(l)}. \quad (20)$$

可仅对部分层集合  $\mathcal{L} \subseteq \{1, \dots, L\}$  开启注入，以在情绪强度与音质稳定性之间取得更好的折中。

### 3 数据集描述

#### 3.1 用于训练模型的数据

本实验训练阶段采用论文提出的 **EmoVoice-DB** 英文情感语音数据集。该数据集为高质量合成情感语音语料：首先由大模型生成“文本内容 + 细粒度情绪自然语言描述”的配对标注，再使用具备较强情感可控能力的语音合成系统生成对应语音，并通过转写一致性筛选漏读、误读或偏离脚本的样本，从而在保证情感表达力的同时维持较高的文本忠实度。

**数据规模与情绪覆盖** EmoVoice-DB 共包含 22100 条英文语音样本，约 40.45 小时音频，覆盖 7 类核心情绪：*angry, happy, sad, surprised, fearful, disgusted, neutral*，各类样本数量相对均衡；同时包含 5 种不同说话人音色，以提升音色与表达风格多样性。数据集统计如表 1 所示。

表 1: EmoVoice-DB 数据集统计（按情绪类别）

| Emotion   | Count | Duration (h) |
|-----------|-------|--------------|
| Angry     | 3486  | 5.76         |
| Happy     | 3269  | 6.02         |
| Sad       | 3174  | 6.94         |
| Surprised | 3072  | 5.67         |
| Fearful   | 2961  | 5.52         |
| Disgusted | 2950  | 5.59         |
| Neutral   | 3188  | 4.95         |
| Sum       | 22100 | 40.45        |

**训练/验证/测试划分** 为便于统一评测与复现实验，论文对 EmoVoice-DB 采用按情绪类别分层抽样：每类情绪随机抽取 100 条作为测试集、50 条作为验证集，其余样本作为训练集。本实验遵循该划分方式使用训练部分数据。

**单条样本结构与字段含义** 本实验代码使用的训练样本以 JSON 形式组织，既包含“粗粒度情绪标签”，也包含“细粒度情绪描述文本”，并提供与离散建模直接相关的语音 token 标注。样例字段含义如下：

- `key`: 样本唯一标识符，通常编码样本编号、情绪类别及说话人音色等信息，便于索引与追踪。
- `source_text`: 输入端文本（待朗读文本）。训练时作为内容约束，要求语音逐字对应。
- `target_text`: 目标转写文本，用于监督学习时的内容一致性约束与对齐校验；在本数据中通常与 `source_text` 保持一致。
- `emotion`: 粗粒度情绪类别标签（如 angry、happy 等），用于类别采样、统计与分组评测。
- `emotion_text_prompt`: 细粒度自然语言情绪描述，用于刻画更具体的情绪状态与表达方式（例如愤怒的走向、张力、语气等），相比单一类别标签可提供更丰富的情感控制监督信号。
- `target_wav`: 目标语音波形文件路径，指向该样本的情感语音音频。
- `answer_cosyvoice_speech_token`: 从 `target_wav` 中提取的离散语音语义 token 序列。论文在情感微调数据上提取 50 Hz 的 CosyVoice 语义 token，并将其与文本配对形成训练样本；本实验据此将 TTS 训练目标表述为对离散 token 序列的预测，从而更适配基于 LLM 的自回归框架。

综上，EmoVoice-DB 在每条样本中同时提供文本内容、粗粒度情绪类别与细粒度自然语言情绪描述，并配套离散语音 token 作为学习目标，使模型能够在保证内容一致性的前提下学习更细致、可泛化的情感表达映射。

## 3.2 用于提取情感激活的数据

除使用 EmoVoice-DB 进行情感 TTS 微调外，我们在数据端额外构建了一套用于提取情感激活的配套数据集。该数据集的核心目标是：在文本语义保持一致的前提下，通过对同一文本合成多种情绪版本语音，并为每种情绪提供细粒度情绪描述与连续量化的激活/强度标注，从而为后续的情感激活建模提供稳定的数据支撑。

**数据构建原则** 该数据集按“同文本、多情绪”的结构组织：对每条英文参考文本，在同一音色与相同朗读约束下分别合成 *neutral*、*happy*、*sad* 等多种情绪语音。这样可以最大程度消除文本内容差异带来的干扰，使情绪差异主要体现在声学层面（语速、音高范围、能量、停连、语气张力等），便于提取可泛化的情感激活表示。此外，数据中保留 *style* 与 *voice* 字段，用于记录文本生成风格与说话人音色设定，以覆盖更丰富的语体与音色条件。

**情绪激活标注** 每种情绪除提供自然语言描述 *desc* 外，还包含两类连续标注：

- *arousal*: 情绪激活度（0–1 归一化），反映整体兴奋水平；
- *intensity*: 情绪强度（0–1 归一化），反映情绪表达的用力程度与显著性。

二者共同刻画“情绪类型 + 强度”的连续控制信息，使数据既能支持情绪分类相关分析，也能支持回归式激活建模。

**一致性校验与过滤策略** 为保证“同一文本在不同情绪下逐字朗读”这一前提，我们对每条合成语音进行 ASR 转写并计算与原文本的编辑距离指标（WER 及插入/删除/替换计数）。数据项中记录三种情绪各自的 *asr\_\**、*wer\_\**、*ins/del/sub\_\** 指标，并依据预设阈值策略进行过滤，最终用 *kept* 标记样本是否保留。过滤规则以 *filter\_policy* 字段显式记录，从而兼顾情绪表达自由度与文本一致性。

**单条样本结构说明** 数据以 JSON 组织，示例字段含义如下：

- *id*: 样本编号，用于索引同一条文本的多情绪语音组。
- *style*: 文本写作风格标签（如 *prose*），用于控制文本整体语体与韵律可塑性。
- *voice*: 合成音色/说话人设定（如 *ballad*），同一条样本的多情绪版本保持一致以隔离音色因素。
- *text*: 参考文本内容；该文本在 *neutral/happy/sad* 三个版本中应保持完全一致。
- *neutral/happy/sad*: 三种情绪的结构化标注，每个情绪包含：
  - *desc*: 细粒度情绪描述文本（控制语速、音高、能量、停连等声学表现，不改变语义内容）；
  - *arousal* 与 *intensity*: 连续量化的激活度与强度。
- *wav\_neutral/wav\_happy/wav\_sad*: 三种情绪语音的保存路径。
- *asr\_\*/wer\_\*/ins\_\*/del\_\*/sub\_\**: ASR 校验结果与误差统计，用于过滤与质量分析。

- `kept`: 是否通过过滤并纳入最终用于情感激活提取的数据子集。
- `models`: 生成该条数据使用的 LLM/TTS/ASR 模型名称记录，便于复现实验与溯源分析。
- `filter_policy`: 过滤策略参数（如不同情绪的 WER 上限、最大删除数等），保证数据筛选过程可复现。

总体而言，该数据集以“同文本、多情绪、多音色”的结构提供了可控且可校验的情绪表达样本，并通过连续的 `arousal/intensity` 标注将情绪从离散类别扩展到可回归的激活空间，为后续情感激活提取与建模提供了直接监督与稳定评测依据。

## 4 实验

### 4.1 训练设置

本实验采用监督微调的方式在预训练权重基础上继续训练情感语音合成模型。训练使用 4 张 80GB 显存的 Nvidia H00 显卡。

**模型** 实验的模型架构与 EmoVoice 一致，使用 Qwen2.5-0.5B 为语言模型，输入情感描述与目标文本，生成语音 Tokens，使用 CosyVoice 的语音解码器将生产的 Token 解码为语音。实验中设置语义码层数 `code_layer=3`，在生成音频 token 前引入 5 个 latency tokens；关闭 layer-shift，即不同层使用一致的 codebook 索引空间。解码端启用分组解码，并使用线性适配器进行映射。

**数据与情感条件** 训练集与验证集分别来自 EmoVoice-DB，此外，本实验“数据端”生成的数据经过 CosyVoice 处理后生成语音 Token，也加入了训练集中。

**优化与训练日程** 训练的关键超参数设置如下：全局批次大小为 32，梯度累积步数为 1，采用混合精度训练。训练总轮数为 40，最大学习率设为  $1 \times 10^{-5}$ ，并使用 1000 步 warmup；计划总步数为 78920。参数冻结策略方面，编码器部分被冻结，而语言模型骨干参与训练。训练采用自定义 batching 策略以适配语音 token 序列的长度与结构特性。

**损失函数与对齐约束** 除主训练目标外，本实验启用情感对齐辅助约束，对齐所需的风格嵌入来自预先提取的 `style_emb_pt`。对齐项权重设为 0.5，温度系数为 0.07，并启用全局负样本以增强判别性。本实验训练的交叉熵损失曲线以及对比学习损失曲线如图 4 所示。

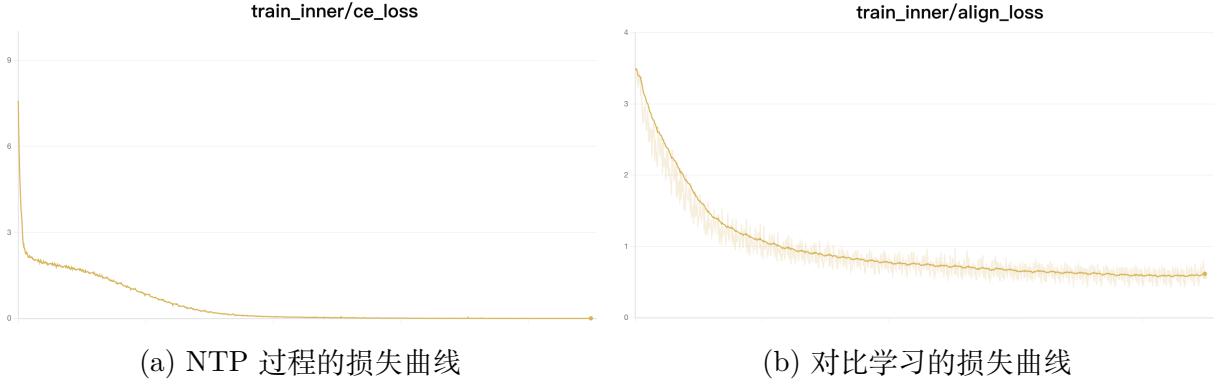


图 4: 联合训练过程中的损失曲线

表 2: 加入对比学习联合训练后的模型在测试集结果

| Method      | WER↓        | emo2vec↑     | Acc↑        | R <sub>angry</sub> ↑ | R <sub>happy</sub> ↑ | R <sub>neutral</sub> ↑ | R <sub>other</sub> ↑ | R <sub>sad</sub> ↑ | AvgRecall↑  |
|-------------|-------------|--------------|-------------|----------------------|----------------------|------------------------|----------------------|--------------------|-------------|
| CosyVoice   | 12.76       | 89.89        | 25.7        | 17.0                 | 53.0                 | 84.0                   | 3.0                  | 13.0               | 34.0        |
| CosyVoice2  | 12.37       | 87.47        | 26.3        | 17.0                 | 55.0                 | 85.0                   | 3.0                  | 14.0               | 34.8        |
| EmoVoice    | 11.32       | 90.40        | 26.9        | 19.0                 | <b>57.0</b>          | 84.0                   | 4.0                  | 16.0               | 36.0        |
| <b>Ours</b> | <b>9.74</b> | <b>92.30</b> | <b>28.0</b> | <b>23.0</b>          | 56.0                 | <b>91.0</b>            | <b>13.0</b>          | <b>22.0</b>        | <b>41.0</b> |

说明: WER 为 ASR 转写词错误率 ( $\downarrow$  越低越好); emo2vec 为基于 emotion2vec 的情感一致性分数 ( $\uparrow$  越高越好); Acc 为情感分类准确率; R 为各情绪类别召回率, AvgRecall 为各类召回率的平均值。

## 4.2 实验结果

在测试集上, 我们从内容一致性与情感可辨识度两个维度评估模型性能, 结果如表 2 所示。内容一致性采用 ASR 转写后的词错误率; 情感相关指标基于 emotion2vec 表征计算, 包括情感相似度分数、情感分类准确率以及按类别的召回率和平均召回。

从整体结果来看, 我们的方法在多数指标上取得最佳表现。首先, 在内容一致性方面, **Ours** 的 WER 为 **9.74**, 显著低于 CosyVoice (12.76)、CosyVoice2 (12.37) 与 EmoVoice (11.32), 说明在引入情感建模与对齐约束后, 模型仍能更稳定地跟随文本逐字生成, 且可懂度更高。其次, 在情感一致性方面, **Ours** 的 emo2vec 达到 **92.30**, 相比 EmoVoice (90.40) 进一步提升, 表明联合训练使生成语音在情感表征空间中更贴近目标情绪。

在情感分类与召回指标上, **Ours** 的 Acc 提升至 **28.0**, 并取得最高的 AvgRecall (**41.0**)。值得注意的是, 我们的方法在 angry、neutral、sad、other 四类召回率上均优于对比方法, 其中 R<sub>neutral</sub> 达到 **91.0**, R<sub>other</sub> 提升至 **13.0**, 显示对比学习式联合训练能够增强类别间区分度, 尤其对易混淆、长尾类别 (如 other) 更有效。虽然 R<sub>happy</sub> 略低于 EmoVoice 的 57.0 (我们为 56.0), 但整体平均召回与多类覆盖显著更优, 说明模型在保证主流情感表现力的同时, 提高了更广泛情绪类别的可辨识性与鲁棒性。总体而言, 表

表 3: 注入情感激活值后的测试集结果 (只测试 happy+sad)

| Method          | WER ↓      | emo2vec ↑   | Acc ↑      | R <sub>happy</sub> ↑ | R <sub>sad</sub> ↑ |
|-----------------|------------|-------------|------------|----------------------|--------------------|
| Non-Inject      | 10.31      | 91.4        | 39.0       | 56.0                 | 22.0               |
| Random-Inject   | 10.86      | 90.7        | 38.0       | 56.0                 | 20.0               |
| Activate-Inject | 9.46(0.85) | 92.73(1.33) | 57.5(18.5) | 77.0(21.0)           | 38.0(16.0)         |

**说明:** 仅在 happy 与 sad 子集评测; 括号内为相对 Non-Inject 的绝对提升 (WER 为绝对下降)。

2 验证了加入对比学习对齐项后, 模型实现了“更低 WER + 更高情感一致性与更均衡召回”的统一改进。

### 4.3 消融实验

为进一步验证“情感激活值注入”在推理阶段的有效性, 我们设计了消融对比实验, 仅在 happy 与 sad 子集上测试, 结果见表 3。我们比较三种设置: **Non-Inject** (不注入激活值)、**Random-Inject** (注入随机向量, 作为控制变量) 以及 **Activate-Inject** (注入从情感激活中提取的定向向量)。表中括号数值表示相对于 Non-Inject 的明显提升。

结果显示, **Random-Inject** 并未带来增益, 反而在 WER (10.86 与 10.31) 与 emo2vec (90.7 与 91.4) 上出现退化, 说明“随意注入扰动”并不能稳定改善情感表达, 甚至会破坏生成稳定性。相对地, **Activate-Inject** 在所有指标上均显著优于 Non-Inject: WER 降至 9.46, emo2vec 提升至 92.73, Acc 从 39.0 大幅提升到 57.5, 并且 R<sub>happy</sub> 与 R<sub>sad</sub> 分别提升 21.0 与 16.0。该结果表明注入的激活向量具有明确的情感方向性, 能够在不牺牲内容一致性的前提下显著增强目标情绪的可控性与可辨识度; 同时 Random-Inject 的对照进一步排除了“仅因额外扰动导致分数提升”的可能性。

## 5 总结与展望

**总结:** 本实验围绕情感语音生成的可控性与一致性, 从数据端、训练端与推理端构建了完整方案并进行了验证。数据端采用“生成—ASR 回转写校验—筛选”的流程, 得到同文本多情绪的高一致性数据, 显式降低漏读与改写等内容噪声。训练端在交叉熵目标上引入对比学习式的情感对齐约束, 使情绪描述与语音情绪表征在隐空间更一致, 从而提升情绪可辨识度与整体稳定性。推理端进一步加入无需再训练的激活引导, 通过在解码器关键层注入可控方向实现情绪增强或削弱, 实验现象表明定向注入相比随机扰动更能稳定提高可控性且对内容一致性影响更小。

**展望:** 后续可从三方面继续改进: (1) 扩展更丰富的情绪类别、说话人与风格, 并引入更全面的自动质检指标以提升数据可靠性; (2) 强化文本情感表征与对齐策略 (如

更有效的池化、难负样本或跨说话人对齐)，提升细粒度描述的泛化能力；(3) 研究激活引导的自适应注入（如按时间步动态调节幅度），在情绪强度与音质稳定之间取得更优平衡，同时补充 MOS 与主观情感可感知度等评测以形成更完整的评价体系。