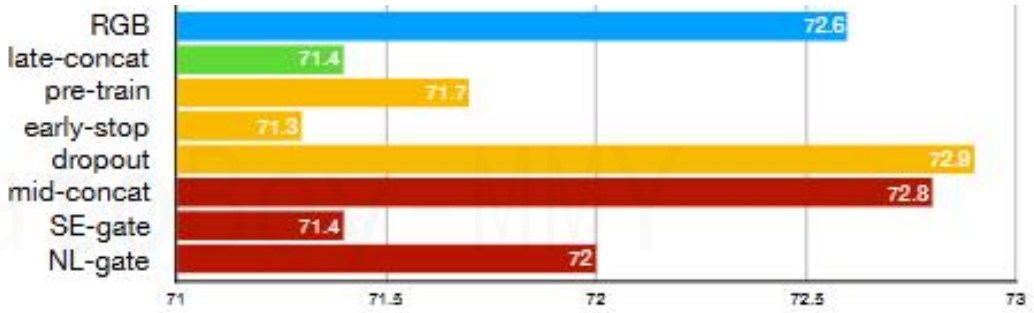
MMPareto:

Boosting Multimodal Learning with Innocent Unimodal Assistance

1 研究背景
2 问题发现
3 数学建模
4 算法设计与代码实现
5 测试结果
6 研究结论

研究背景

多模态学习被提出的初衷是为了能够使得AI像人一样同时接受并学习多种模态的信息，从而提高AI的能力，但是事实却于此相反



Wang, Weiyao, Du Tran, and Matt Feiszli.
"What makes training multi-modal classification networks hard?."
Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

研究背景

多模态学习中常常存在不平衡问题，各个模态之间的利用率无法平衡，使得并非所有模态的“潜力”都能被完整释放，特别是有些模态处于劣势，常常被优势模态压制，使得其中的内容无法被学习。于是学术界提出了类多任务的多模态框架，使用针对单一模态的训练来缓解某一模态被压制的情况

但是此时便会出现另一个问题，
单模态的优化方向可能和多模态
的优化方向不同从而对多模态的
能力产生损害

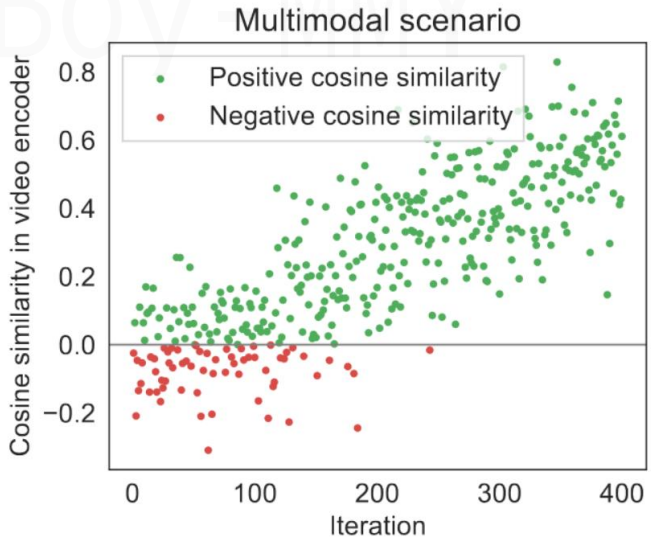


图1　(a)

研究背景

解决融合多模态优化与单模态优化问题的关键就在于：

如何保证能够在不损害多模态训练的前提下使用单模态进行辅助，使最终优化朝着一个对所有目标都有利的方向进行，最后收敛到一个权衡状态。

这种利用学习多个单模态来优化多模态进行联合训练的方法类似于多任务学习，被称为类多任务的多模态框架，于是借鉴了多任务学习的方法：

Pareto方法

帮助多模态学习训练。

问题发现

但是将Pareto算法应用到多模态训练上但却并没有取得良好效果，反而产生了负面影响，与预期不符，本文所研究的核心问题出现了

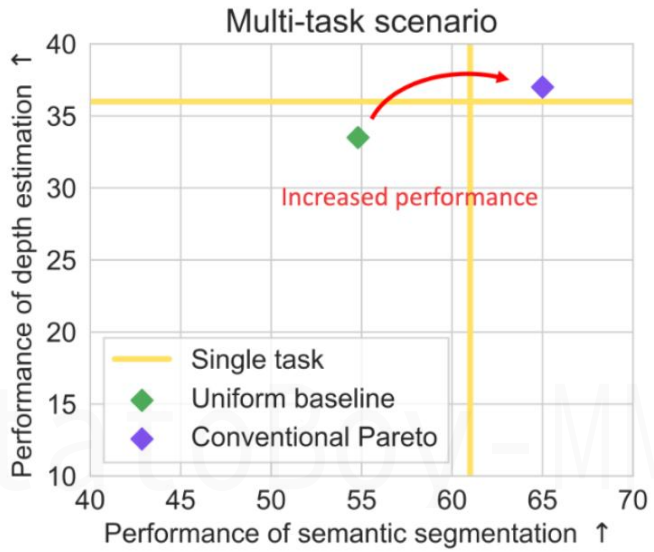


图1b
多任务使用Pareto方法

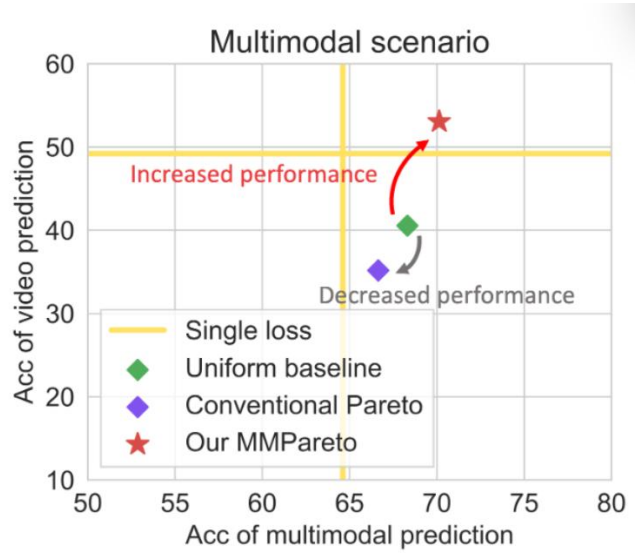


图1c
多模态使用Pareto方法

问题发现

- 什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果

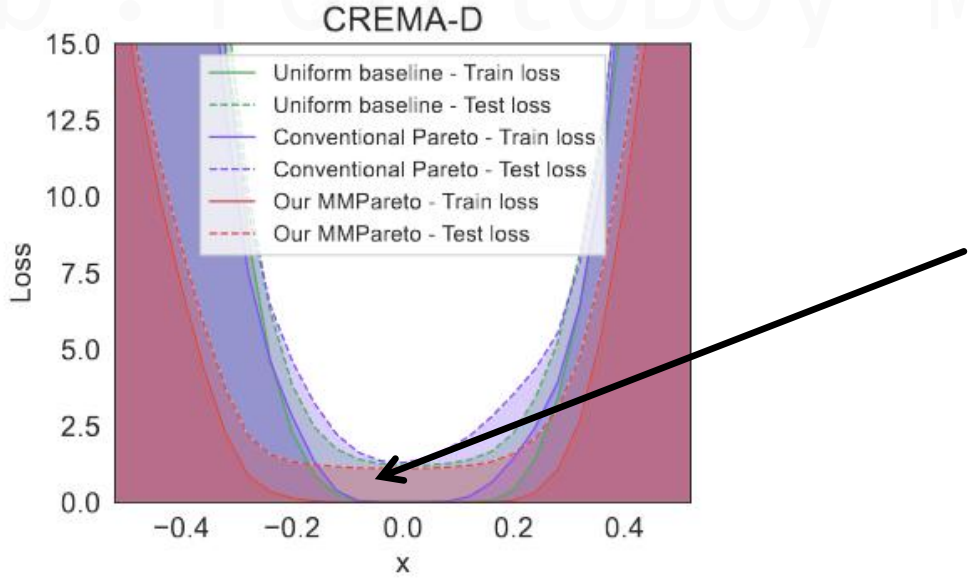
- 如何解决这个问题

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

结论：

传统的Pareto会影响SGD的噪声强度，将模型带到一个更尖锐的极小值，从而削弱了模型的泛化能力



由此图可以直观的看出Pareto的极小值较尖锐

图5（a)

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

前提条件

$$\mathcal{L} = \mathcal{L}_m + \sum_{k=1}^{n} \mathcal{L}_u^k$$

类多任务的多模态框架

$$g_S\big(\theta(t)\big) \sim \mathcal{N}\big(g_N\big(\theta(t)\big),\ \tfrac{1}{|S|}\,C\big)$$

$$g_S^m\big(\theta^k(t)\big) \sim \mathcal{N}\big(g_N^m\big(\theta^k(t)\big),\ \tfrac{1}{|S|}\,C^m\big)$$

$$g_S^u\big(\theta^k(t)\big) \sim \mathcal{N}\big(g_N^u\big(\theta^k(t)\big),\ \tfrac{1}{|S|}\,C^u\big)$$

多模态联合训练及单模态训练的梯度分布

$$\min_{\alpha^m,\,\alpha^u \in \mathbb{R}} \left\| \alpha^m g_S^m + \alpha^u g_S^u \right\|^2 \quad \text{s.t.} \quad \alpha^m,\, \alpha^u \geq 0, \quad \alpha^m + \alpha^u = 1,$$

Pareto所要解决的问题

数学建模

- 什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果

前提条件

$$g_S^m\left(\theta^k(t)\right) \sim \mathcal{N}\left(g_N^m\left(\theta^k(t)\right), \frac{1}{|S|} C^m\right)$$

$$g_S^u\left(\theta^k(t)\right) \sim \mathcal{N}\left(g_N^u\left(\theta^k(t)\right), \frac{1}{|S|} C^u\right)$$

假设

$$c^u > c^m$$

多模态的不确定性更小

$$g_N^u(\theta^k(t)) > g_N^m(\theta^k(t))$$

多模态的幅度更小

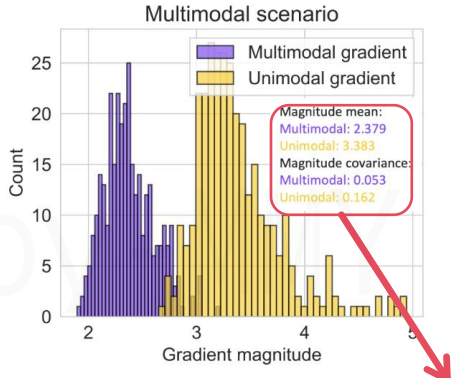


图1（d）

Magnitude mean:
Multimodal: 2.379
Unimodal: 3.383
Magnitude covariance:
Multimodal: 0.053
Unimodal: 0.162

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

数学推导

求解 $\min_{\alpha^m,\,\alpha^u\in\mathbb{R}} \left\|\alpha^m g_S^m+\alpha^u g_S^u\right\|^2 \quad \text{s.t.} \quad \alpha^m,\,\alpha^u\geq 0, \quad \alpha^m+\alpha^u=1,$

首先

$$\begin{aligned}\alpha_m &= \alpha\\ \alpha_m &= 1-\alpha\\ g_s^m &= g_m\\ g_s^u &= g_u\end{aligned}$$

$$\begin{aligned}f(\alpha) &= \|\alpha g_m+(1-\alpha)g_u\|^2\\ &= \|g_m+\alpha(g_m-g_u)\|^2\\ &= \|g_m\|^2+2\alpha(g_m-g_u)^T g_u+\alpha^2\|g_m-g_u\|^2\end{aligned}$$

求导 $f'(\alpha)=2(g_m-g_u)^T g_u+2\alpha\parallel g_m-g_u\parallel^2=0$

求解 $\alpha=\dfrac{(g_u-g_m)^T g_u}{\|g_m-g_u\|^2}$

范围

$$\alpha\geq 1\Rightarrow\frac{(g_u-g_m)^T g_u}{||g_m-g_u||^2}>1\Rightarrow\cos\beta>\frac{||g_m||}{||g_u||}$$

$$\begin{cases}\alpha^m=1, \quad \alpha^u=0, & \cos\beta\ \geq\ \dfrac{\|g_s^m\|}{\|g_s^u\|},\\ \alpha^m=\dfrac{(g_s^u-g_s^m)^\top g_s^u}{\|g_s^m-g_s^u\|^2}, \quad \alpha^u=1-\alpha^m, & \text{otherwise},\end{cases}$$

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

数学推导

求解 $\min_{\alpha^m,\,\alpha^u\in\mathbb{R}} \left\|\alpha^m g_S^m+\alpha^u g_S^u\right\|^2 \quad \text{s.t.} \quad \alpha^m,\,\alpha^u\geq 0, \quad \alpha^m+\alpha^u=1,$

当 $\cos\beta > \frac{||g_s^m||}{||g_s^u||}$ 时，显然 $\alpha^m > \alpha^u$

其他情况下 $\alpha^m - \alpha^u$

$$
\begin{aligned}
&= \frac{(\mathbf{g}_S^u-\mathbf{g}_S^m)^\top \mathbf{g}_S^u}{\left\|\mathbf{g}_S^m-\mathbf{g}_S^u\right\|^2} - \left(1-\frac{(\mathbf{g}_S^u-\mathbf{g}_S^m)^\top \mathbf{g}_S^u}{\left\|\mathbf{g}_S^m-\mathbf{g}_S^u\right\|^2}\right) \\
&= \frac{\left\|\mathbf{g}_S^u\right\|^2 - \left\|\mathbf{g}_S^u\right\|\left\|\mathbf{g}_S^m\right\|\cos\beta}{\left\|\mathbf{g}_S^m-\mathbf{g}_S^u\right\|^2} - \frac{\left\|\mathbf{g}_S^m\right\|^2 - \left\|\mathbf{g}_S^u\right\|\left\|\mathbf{g}_S^m\right\|\cos\beta}{\left\|\mathbf{g}_S^m-\mathbf{g}_S^u\right\|^2} \\
&> 0. \quad (\left\|\mathbf{g}_S^m\right\| < \left\|\mathbf{g}_S^u\right\|)
\end{aligned}
$$

可以看出多模态梯度得到的权重是更大的

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

数学推导

无任何梯度集成策略 $h_S\left(\theta^k(t)\right) \sim \mathcal{N}\left(g_N^m\left(\theta^k(t)\right)+g_N^u\left(\theta^k(t)\right), \frac{C^m+C^u}{|S|}\right)$

Pareto梯度集成策略 $\mathbf{h}_S^{\text{Pareto}}\left(\theta^k(t)\right) \sim \mathcal{N}\left(\mathbf{h}_N^{\text{Pareto}}\left(\theta^k(t)\right), \frac{(2\alpha^m)^2C^m+(2\alpha^u)^2C^u}{|S|}\right)$

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

数学推导

$$\theta^k(t+1) = \theta^k(t) - \eta \mathbf{h}_S(\theta^k(t)),$$
$$= \theta^k(t) - \eta \mathbf{h}_N(\theta^k(t)) + \eta\epsilon_t,$$

这里的 $\epsilon_t \sim \mathcal{N}\left(0, \frac{C^m + C^u}{|S|}\right)$ 的即为噪声项

同样的可以得到Pareto的噪声项为

$$\zeta_t \sim \mathcal{N}\left(0, \frac{(2\alpha^m)^2 C^m + (2\alpha^u)^2 C^u}{|S|}\right)$$

$$k\,C^m = C^u$$

$$(2\alpha^m)^2 C^m + (2\alpha^u)^2 C^u < C^m + C^u$$

$$(2\alpha^m)^2 C^m + \left(2(1-\alpha^m)\right)^2 k C^m < (k+1)C^m$$

$$(2\alpha^m)^2 + 4k - 8\alpha^m k + (2\alpha^m)^2 k < k+1$$

$$\left(\alpha^m - \tfrac{1}{2}\right)\left((4k+4)\alpha^m + 2 - 6k\right) < 0$$

$$\frac{1}{2} < \alpha^m < \frac{3k-1}{2k+2}$$

当k大于3时，Patero集成的噪声必定小于无梯度集成策略的噪声造成泛化的损害，若k小于等于3时，则当多模态权重在这个范围内时，Patero集成的噪声也小于无梯度集成策略的噪声造成泛化的损害

数学建模

- **什么原因导致了Pareto方法应用到多模态学习之中就无法达到预期效果**

数学推导

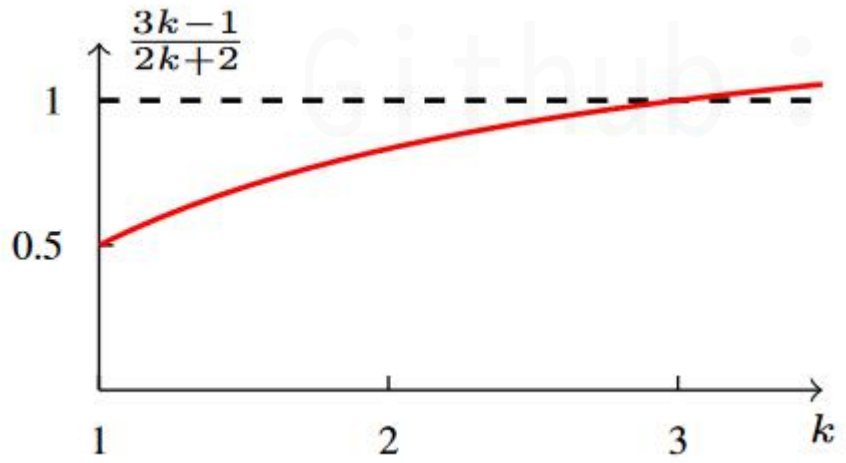


图3

从图三来看，在k小于等于3时，多模态权重也是大概率是落到这个范围内。
由以上所有的证明来看在训练过程中的大多数情况下，使用Pareto必定造成模型的损害

数学建模

- **如何解决这个问题**

在保留Pareto能够集成各个梯度的幅度和方向的思想下，将噪声增大，提高模型的**泛化**能力

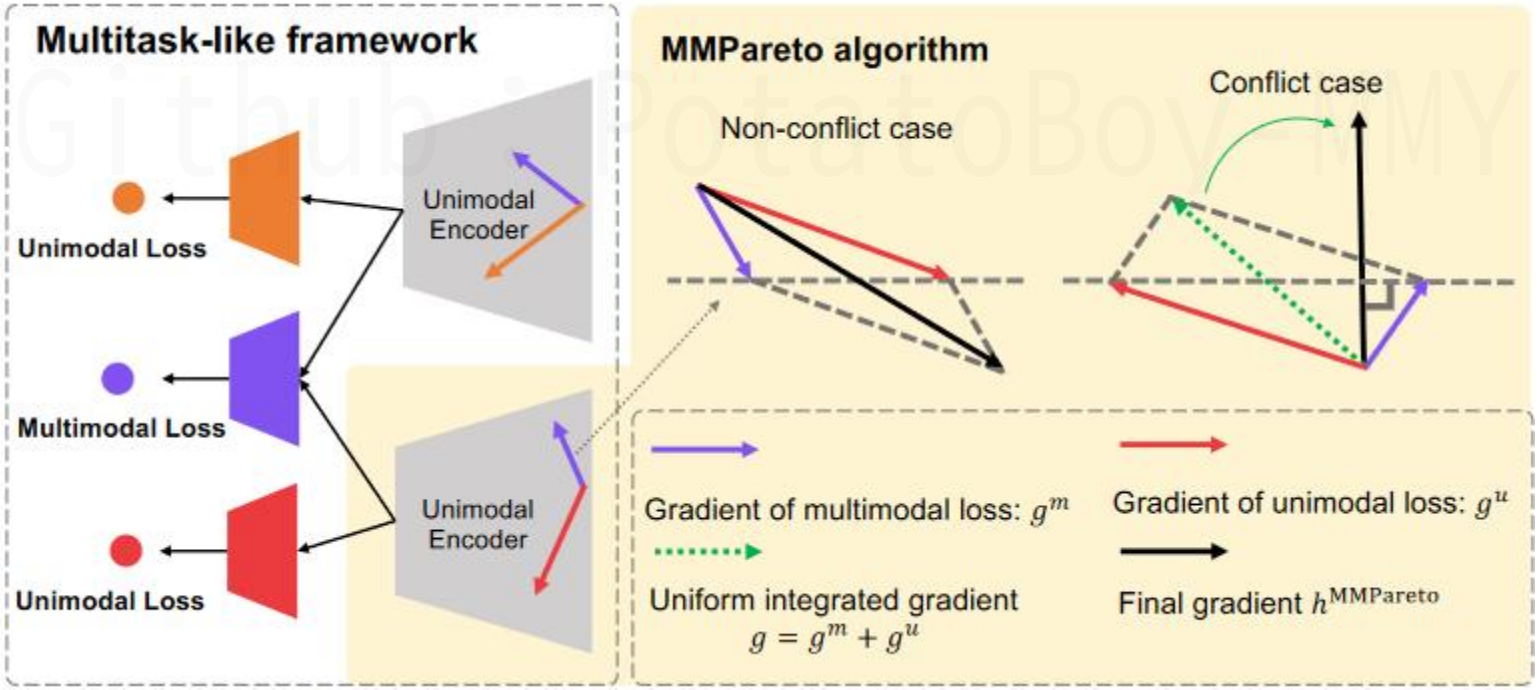


图2 MMPareto的整体框架图

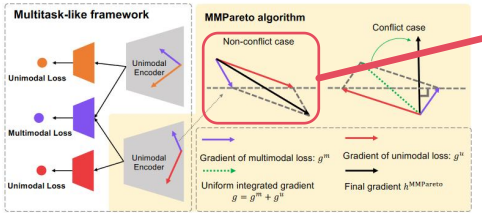
数学建模

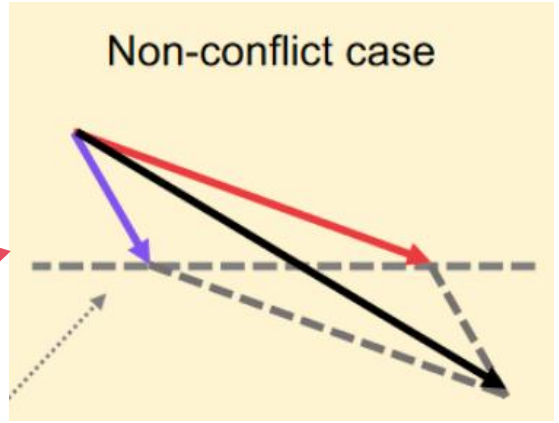
- **如何解决这个问题**

非冲突

直接将两个权重都设为1，即直接采用基础融合方案，既可以实现方向和梯度的融合，也可以保证噪音不受损

$$\mathbf{h}_S^{\text{MMPareto}} \sim \mathcal{N}\left(\mathbf{g}_N^m + \mathbf{g}_N^u, \frac{C^m + C^u}{|S|}\right)$$





数学建模

- **如何解决这个问题**

冲突

Patero的方向找到的方向是没有错的，但是直接用这个权重是会损害噪声，那我们就是用这个方向，但是不使用Pareto得到的权重大小

Pareto得到的方向 $\dfrac{2\alpha^m \mathbf{g}_S^m + 2\alpha^u \mathbf{g}_S^u}{\|2\alpha^m \mathbf{g}_S^m + 2\alpha^u \mathbf{g}_S^u\|}$

不损害噪音的大小 $\|\mathbf{g}_S^m + \mathbf{g}_S^u\|$

$$\mathbf{h}_S^{\text{MMPareto}} \sim \mathcal{N}\left(\lambda \mathbf{h}_N^{\text{Pareto}},\ \lambda^2 \frac{(2\alpha^m)^2 C^m + (2\alpha^u)^2 C^u}{|S|}\right)$$

数学建模

- **如何解决这个问题**

冲突

$$\mathbf{h}_S^{\text{MMPareto}} \sim \mathcal{N}\left(\lambda\,\mathbf{h}_N^{\text{Pareto}},\ \lambda^2\,\frac{(2\alpha^m)^2 C^m + (2\alpha^u)^2 C^u}{|S|}\right)$$

$$\mathbf{h}_S^{\text{MMPareto}} = \frac{2\alpha^m\,\mathbf{g}_S^m + 2\alpha^u\,\mathbf{g}_S^u}{\left\|2\alpha^m\,\mathbf{g}_S^m + 2\alpha^u\,\mathbf{g}_S^u\right\|} \cdot \left\|\mathbf{g}_S^m + \mathbf{g}_S^u\right\| \longrightarrow \lambda = \frac{\left\|\mathbf{g}_S^m + \mathbf{g}_S^u\right\|}{\left\|2\alpha^m\,\mathbf{g}_S^m + 2\alpha^u\,\mathbf{g}_S^u\right\|} > 1$$

由此可以看出MMPareto的噪声强度大于传统Pareto的噪声强度，进一步增强

$$\mathbf{h}_S^{\text{MMPareto}} = \gamma\,\mathbf{h}_S^{\text{MMPareto}}$$

算法设计与代码实现

Algorithm 1 MMPareto

Require: Training dataset $\mathcal{D}$, iteration number T, initialized unimodal encoder parameters θ^k, $k \in \{1, 2, \cdots, n\}$, other parameters θ^{other}.

for $t = 0, \cdots, T-1$ **do**
  Sample a fresh mini-batch S from $\mathcal{D}$;
  Feed-forward the batched data S to the model;
  Calculate gradient using back-propagation;
  Update θ^{other} without gradient integration method;
  for $k = 1, \cdots, n$ **do**
    Obtain $\mathbf{g}_S^m$ and $\mathbf{g}_S^u$ for k-th unimodal encoder;
    Calculate $\cos\beta$; β is angle between $\mathbf{g}_S^m$ and $\mathbf{g}_S^u$;
    Solve problem of Equation 5, obtain α^m, α^u;
    if $\|\alpha^m \mathbf{g}_S^m + \alpha^u \mathbf{g}_S^u\| = 0$ **then**
      Find the Pareto stationarity;
    end if
    if $\cos\beta \geq 0$ **then**
      $2\alpha^m = 2\alpha^u = 1$;
    end if
    Integrate gradient: $\mathbf{h}_S' = 2\alpha^m \mathbf{g}_S^m + 2\alpha^u \mathbf{g}_S^u$;
    $\mathbf{h}_S^{\text{MMPareto}} = \underbrace{\mathbf{h}_S'/\|\mathbf{h}_S'\|}_{\text{Keep non-conflict direction}} \cdot \underbrace{\gamma\|\mathbf{g}_S^m + \mathbf{g}_S^u\|}_{\text{Enhanced magnitude}}$;
    Update θ^k with $\mathbf{h}_S^{\text{MMPareto}}$.
  end for
end for

已经到达Pareto的平稳状态

非冲突状态，直接使用基础融合

使用MMPareto的融合方案

算法设计与代码实现

min_norm_solvers.py

```python
def _min_norm_element_from2(v1v1, v1v2, v2v2):  3 用法  echo0409
    """
    Analytical solution for min_{c} |cx_1 + (1-c)x_2|_2^2
    d is the distance (objective) optimzed
    v1v1 = <x1,x1>
    v1v2 = <x1,x2>
    v2v2 = <x2,x2>
    """
    if v1v2 >= v1v1:
        # Case: Fig 1, third column
        gamma = 0.999
        cost = v1v1
        return gamma, cost
    if v1v2 >= v2v2:
        # Case: Fig 1, first column
        gamma = 0.001
        cost = v2v2
        return gamma, cost
    # Case: Fig 1, second column
    gamma = -1.0 * ( (v1v2 - v2v2) / (v1v1+v2v2 - 2*v1v2) )
    cost = v2v2 + gamma*(v1v2 - v2v2)
    return gamma, cost
```

解出解析解

$$\begin{cases} \alpha^m = 1, \quad \alpha^u = 0, & \cos\beta \geq \dfrac{\|g_s^m\|}{\|g_s^u\|}, \\ \alpha^m = \dfrac{(g_s^u - g_s^m)^\top g_s^u}{\|g_s^m - g_s^u\|^2}, \quad \alpha^u = 1 - \alpha^m, & \text{otherwise}, \end{cases}$$

算法设计与代码实现

min_norm_solvers.py

```python
def _min_norm_2d(vecs, dps):  2 用法  echo0409
    """
    Find the minimum norm solution as combination of two points
    This is correct only in 2D
    ie. min_c |\sum c_i x_i|_2^2 st. \sum c_i = 1 , 1 >= c_1 >= 0 for all i, c_i + c_j = 1.0 
    """
    dmin = 1e8
    for i in range(len(vecs)):
        for j in range(i+1,len(vecs)):
            if (i,j) not in dps:
                dps[(i, j)] = 0.0
                for k in range(len(vecs[i])):
                    dps[(i,j)] += torch.mul(vecs[i][k], vecs[j][k]).sum().data.cpu()
                dps[(j, i)] = dps[(i, j)]
            if (i,i) not in dps:
                dps[(i, i)] = 0.0
                for k in range(len(vecs[i])):
                    dps[(i,i)] += torch.mul(vecs[i][k], vecs[i][k]).sum().data.cpu()
            if (j,j) not in dps:
                dps[(j, j)] = 0.0
                for k in range(len(vecs[i])):
                    dps[(j, j)] += torch.mul(vecs[j][k], vecs[j][k]).sum().data.cpu()
            c,d = MinNormSolver._min_norm_element_from2(dps[(i,i)], dps[(i,j)], dps[(j,j)])
            if d < dmin:
                dmin = d
                sol = [(i,j),c,d]
    return sol, dps
```

计算出所有的范数和权重

测试结果

Method	CREMA-D			Kinetics Sounds		
	Acc	Acc audio	Acc video	Acc	Acc audio	Acc video
Audio-only	-	61.69	-	-	<u>53.63</u>	-
Video-only	-	-	<u>56.05</u>	-	-	49.20
Unimodal pre-trained & fine-tune	71.51	60.08	**60.22**	68.75	53.49	<u>50.07</u>
One joint loss*	66.13	59.27	36.56	64.61	52.03	35.47
Uniform baseline	71.10	<u>63.44</u>	51.34	68.31	53.20	40.55
G-Blending (Wang et al., 2020)	<u>72.01</u>	60.62 (↓)	52.23	<u>68.90</u>	52.11 (↓)	41.35
OGM (Peng et al., 2022)*	69.19 (↓)	56.99 (↓)	40.05 (↓)	66.79 (↓)	51.09 (↓)	37.86 (↓)
Greedy (Wu et al., 2022)*	67.61 (↓)	60.69 (↓)	38.17 (↓)	65.32 (↓)	50.58 (↓)	35.97 (↓)
PMR (Fan et al., 2023)*	66.32 (↓)	59.95 (↓)	32.53 (↓)	65.70 (↓)	52.47 (↓)	34.52 (↓)
AGM (Li et al., 2023)*	70.06 (↓)	60.38 (↓)	37.54 (↓)	66.17 (↓)	51.31 (↓)	34.83 (↓)
MMPareto	**75.13**	**65.46**	55.24	**70.13**	**56.40**	**53.05**

表1 与不平衡多模态学习方法比较

测试结果

Method	CREMA-D				Kinetics Sounds			
	from scratch		with pretrain		from scratch		with pretrain	
	Acc	macro F1	Acc	macro F1	Acc	macro F1	Acc	macro F1
One joint loss	44.96	42.78	66.69	67.26	42.51	41.56	68.30	69.31
Uniform baseline	45.30	43.74	69.89	70.11	43.31	43.08	69.40	69.60
G-Blending (Wang et al., 2020)	46.38	45.16	69.91	70.01	44.69	44.19	69.41	69.47
OGM-GE (Peng et al., 2022)	42.88	39.34	65.73	65.88	41.79	41.09	69.55	69.53
Greedy (Wu et al., 2022)	44.49	42.76	66.67	67.26	43.31	43.08	69.62	69.75
PMR (Fan et al., 2023)	44.76	42.95	65.59	66.07	43.75	43.21	69.67	69.87
AGM (Li et al., 2023)	45.36	43.81	66.54	67.75	43.65	43.57	69.59	69.14
MMPareto	**48.66**	**48.17**	**70.43**	**71.17**	**45.20**	**45.26**	**70.28**	**70.11**

表2 在Transformer上的表现

测试结果

Method	CG-MNIST		ModelNet40		Kinetics Sounds	
	Acc	macro F1	Acc	macro F1	Acc	macro F1
One joint loss	60.50	59.89	87.88	83.32	64.61	64.12
Uniform baseline	75.68	75.66	89.18	84.69	68.31	68.13
Conventional Pareto (Sener & Koltun, 2018)	62.00	61.85	88.05	83.01	66.64	66.17
GradNorm (Chen et al., 2018)	76.16	76.12	88.98	83.79	65.84	65.14
PCGrad (Yu et al., 2020)	79.35	77.14	89.59	84.44	69.11	68.75
MetaBalance (He et al., 2022)	79.18	77.87	89.63	84.87	68.90	68.62
MMPareto	**81.88**	**81.69**	**89.95**	**85.15**	**70.13**	**70.18**

表3 在多任务上的表现

测试结果

MMPareto的其他好处

Method	Acc	macro F1
One joint loss	75.07	73.16
Uniform baseline	75.95	73.93
G-Blending (Wang et al., 2020)	76.16	74.65
OGM* (Peng et al., 2022)	75.80	74.71
Greedy (Wu et al., 2022)	/	/
PMR* (Fan et al., 2023)	76.28	75.06
AGM (Li et al., 2023)	76.08	74.98
MMPareto	**76.53**	**75.59**

表7 MMPareto可以扩展到多个

Table 8. Time cost of MMPareto calculation per iteration.

Dataset	Time cost mean	Time cost variance
CREMA-D	0.11s	1e-5
Kinetics Sounds	0.16s	1e-5
CG-MNIST	0.05s	1e-5
ModelNet40	0.10s	1e-5

Table 9. Number of iterations that find Pareto stationarity.

Dataset	Audio encoder	Video encoder
CREMA-D	#3098	#1709
Kinetics Sounds	#10197	#7957

表8与表9 MMPareto不会过度增加训练时间，成本可接受

研究结论

MMPareto算法能够融合单模态和多模态梯度，使得两者都能收益且能够保证噪声不变小，模型的泛化能力不受损，且其适用范围广泛，灵活性高，具有较大价值

谢谢！

恳请批评指正