Balanced Multimodal Learning via On-the-fly Gradient Modulation

Presenter: Connor Weeks
Overview

Problem Statement
Existing multimodal models could have under-optimized representations due to another dominant modality.

Contribution
Proposes a novel training procedure which measures the discrepancy between modalities to balance training and improve performance.
Related Work
Multimodal Learning

New modalities can boost performance in
• Action recognition
• Audio-visual speech recognition
• Visual question answering

New modalities can have:
• Different convergence rates
• Information discrepancies

These can bias training toward one modality

video cluster #27, purity: 0.36
Gradient-Blending

• is an approach for improving multimodal balancing.

• “computes an optimal blending of modalities based on their overfitting behaviors.”

• targets overfitting rather than penalizing a dominant modality.
Gradient-Blending

Definition of Overfitting-to-Generalization Ratio (OGR)

\[ OGR \equiv \left| \frac{\Delta O_{N,n}}{\Delta G_{N,n}} \right| = \left| \frac{O_{N+n} - O_N}{\mathcal{L}_{N}^{*} - \mathcal{L}_{N+n}^{*}} \right| \]

Gradient-Blending: computes an optimal blending of multiple gradients to minimize \( OGR \)

Uses small checkpoints to allow each gradient step to be more easily calculated.

Related Work
Gradient-Blending

Related Work

Equation for optimal gradient blend

\[ w^* = \arg \min_w \mathbb{E} \left[ \left( \frac{\langle \nabla \mathcal{L}^T - \nabla \mathcal{L}^*, \sum_k w_k v_k \rangle}{\langle \nabla \mathcal{L}^*, \sum_k w_k v_k \rangle} \right)^2 \right] \]

per-modality weights

\[ w_k^* = \frac{1}{Z} \frac{\langle \nabla \mathcal{L}^* , v_k \rangle}{\sigma_k^2} \]

final loss calculation

\[ \mathcal{L}_{blend} = \sum_{i=1}^{k+1} w_i \mathcal{L}_i \]

Online vs Offline Blending

**Algorithm 2: Offline Gradient-Blending**

- **input:** \( \varphi^0 \), Initialized model
- \( N \), # of epochs

- **Result:** Trained multi-head model \( \varphi^N \)

- Compute per-modality weights

\[ \{w_i\}_{i=1}^k = GB.Estimater(\varphi^0, N) \]

- Train \( \varphi^0 \) with \( \{w_i\}_{i=1}^k \) for \( N \) epochs to get \( \varphi^N \)

Related Work
Gradient-Blending

Results on Kinetics dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Clip</th>
<th>V@1</th>
<th>V@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Training</td>
<td>61.8</td>
<td>71.7</td>
<td>89.6</td>
</tr>
<tr>
<td>RGB Only</td>
<td>63.5</td>
<td>72.6</td>
<td>90.1</td>
</tr>
<tr>
<td>Offline G-Blend</td>
<td>65.9</td>
<td>74.7</td>
<td>91.5</td>
</tr>
<tr>
<td>Online G-Blend</td>
<td><strong>66.9</strong></td>
<td><strong>75.8</strong></td>
<td><strong>91.9</strong></td>
</tr>
</tbody>
</table>

Results across 3 modalities: RGB image, Audio, Optical Flow.

<table>
<thead>
<tr>
<th>Modal</th>
<th>RGB + A</th>
<th>RGB + OF</th>
<th>OF + A</th>
<th>RGB + OF + A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weights</td>
<td>[RGB,A,Join]=[0.630,0.014,0.356]</td>
<td>[RGB,OF,Join]=[0.309,0.495,0.196]</td>
<td>[OF,A,Join]=[0.827,0.011,0.162]</td>
<td>[RGB,OF,A,Join]=[0.33,0.53,0.01,0.13]</td>
</tr>
<tr>
<td>Metric</td>
<td>Clip</td>
<td>V@1</td>
<td>V@5</td>
<td>Clip</td>
</tr>
<tr>
<td>Uni</td>
<td>63.5</td>
<td>72.6</td>
<td>90.1</td>
<td>63.5</td>
</tr>
<tr>
<td>Naive</td>
<td>61.8</td>
<td>71.4</td>
<td>89.3</td>
<td>62.2</td>
</tr>
<tr>
<td>G-Blend</td>
<td><strong>65.9</strong></td>
<td><strong>74.7</strong></td>
<td><strong>91.5</strong></td>
<td><strong>64.3</strong></td>
</tr>
</tbody>
</table>

Related Work

Settings
- Audio represented with log-Mel
- RGB encoder is ResNet3D-based
- Modalities fused with two-layer FC-layer
- SGD optimizer
Gradient-Blending

- Some accuracy drops compared to single-modality predictions
- Achieves SotA results on:
  - Kinetics
  - Sports1M
  - AudioSet
- Monitors overfitting separately for each modality

Related Work
Other Related Methods

**Improving Multimodal Learning with Uni-modal Teachers.**
Proposes the Uni-Modal Teacher (UMT) method to combine uni-modal knowledge. Separate networks for each modality, then are used as teachers to distill a multimodal model.

**Learning to Balance the Learning Rates Between Various Modalities via Adaptive Tracking Factor**
Defines an adaptive tracking factor (ATF) to adjust the learning rate of each modality. Proposes other methods to update the ATF, avoiding unimodal overfitting or underfitting.
Methodology (OGM)
Overview of Methodology

Component 1:
On-the-Fly Gradient Modulation (OGM)
Determines the relative balance for learning each modality.

Component 2:
Generalization Enhancement (GE)
Adds Gaussian noise to gradients to increase generalizability.
On-the-fly Gradient Modulation (OGM)

Eq. 2) \[ f(x_i) = W^a \cdot \varphi^a(\theta^a, x_i^a) + W^v \cdot \varphi^v(\theta^v, x_i^v) + b. \]

Full network

audio encoder

image encoder

Eq. 6) \[ \theta^u_{t+1} = \theta^u_t - \eta \nabla_{\theta^u} L(\theta^u_t). \]

General equation for gradient descent

Eq. 7) \[ \theta^u_{t+1} = \theta^u_t - \eta \tilde{g}(\theta^u_t), \]

For stochastic gradient descent (SGD)
On-the-fly Gradient Modulation (OGM)

\[
\text{Eq. 2)} \quad f(x_i) = W^a \cdot \varphi^a(\theta^a, x_i^a) + W^v \cdot \varphi^v(\theta^v, x_i^v) + b.
\]

Full network

## Methodology

\[
\text{Eq. 6)} \quad \theta_{t+1}^u = \theta_t^u - \eta \nabla \theta_u L(\theta_t^u).
\]

General equation for gradient descent

\[
\text{Eq. 7)} \quad \theta_{t+1}^u = \theta_t^u - \eta \tilde{g}(\theta_t^u),
\]

For stochastic gradient descent (SGD)
On-the-fly Gradient Modulation (OGM)

\[ f(x_i) = W^a \cdot \varphi^a(\theta^a, x_i^a) + W^v \cdot \varphi^v(\theta^v, x_i^v) + b. \]

Eq. 2

For stochastic gradient descent (SGD)

\[ \theta^u_{t+1} = \theta^u_t - \eta \nabla_{\theta^u} L(\theta^u_t). \]

Eq. 6

General equation for gradient descent

\[ \theta^u_{t+1} = \theta^u_t - \eta \tilde{g}(\theta^u_t), \]

Eq. 7
On-the-fly Gradient Modulation (OGM)

Eq. 8)

\[
s^a_i = \sum_{k=1}^{M} 1_{k=y_i} \cdot \text{softmax}(W^a_t \cdot \varphi^a_t(\theta^a, x^a_i) + \frac{b}{2})_k, \quad \text{audio performance}
\]

Approximation of performance

Eq. 9)

\[
\rho^v_t = \frac{\sum_{i \in B_t} s^v_i}{\sum_{i \in B_t} s^a_i},
\]

discrepancy ratio

over minibatch
On-the-fly Gradient Modulation (OGM)

Methodology

\[ s_{i}^{a} = \sum_{k=1}^{M} 1_{k=y_{i}} \cdot \text{softmax}(W_{t}^{a} \cdot \varphi_{t}^{a}(\theta^{a}, x_{i}^{a}) + \frac{b}{2})_{k}, \quad \text{audio performance} \]

\[ s_{i}^{v} = \sum_{k=1}^{M} 1_{k=y_{i}} \cdot \text{softmax}(W_{t}^{v} \cdot \varphi_{t}^{v}(\theta^{v}, x_{i}^{v}) + \frac{b}{2})_{k}, \quad \text{image performance} \]

Approximation of performance

\[ \rho_{t}^{v} = \frac{\sum_{i \in B_{t}} s_{i}^{v}}{\sum_{i \in B_{t}} s_{i}^{a}}. \]

discrepancy ratio

Eq. 8)

Eq. 9)
On-the-fly Gradient Modulation (OGM)

OGM-GE creates a noticeable drop in discrepancy ratio.

Eq. 9) \[
\rho_t^v = \frac{\sum_{i \in B_t} s_i^v}{\sum_{i \in B_t} s_i^a}.
\]

discrepancy ratio

over minibatch
On-the-fly Gradient Modulation (OGM)

\[
\begin{align*}
\text{Eq. 10) } \quad k^u_t &= \begin{cases} 
1 - \tanh(\alpha \cdot \rho^u_t) & \rho^u_t > 1 \\
1 & \text{others,}
\end{cases} \\
\text{Without OGM} &\quad \theta^u_{t+1} = \theta^u_t - \eta \tilde{g}(\theta^u_t), \\
\text{With OGM} &\quad \theta^u_{t+1} = \theta^u_t - \eta \cdot k^u_t \tilde{g}(\theta^u_t).
\end{align*}
\]

Note: the minimum possible penalty is:
\[1 - \tanh(\alpha)\]
On-the-fly Gradient Modulation (OGM)

Eq. 10) \( k_t^u = \begin{cases} 1 - \tanh(\alpha \cdot \rho_t^u) & \rho_t^u > 1 \\ 1 & \text{others,} \end{cases} \)

Without OGM

Eq. 7) \( \theta_{t+1}^u = \theta_t^u - \eta \tilde{g}(\theta_t^u), \)

With OGM

Eq. 11) \( \theta_{t+1}^u = \theta_t^u - \eta \cdot k_t^u \tilde{g}(\theta_t^u). \)

Note: the minimum possible penalty is: \( 1 - \tanh(\alpha) \)
Methodology (GE)
Generalization Enhancement (GE)

The gradient follows a normal distribution as shown by the Central Limit Theorem

\[ \tilde{g}(\theta_t^u) \sim \mathcal{N}(\nabla_{\theta^u} L(\theta_t^u), \Sigma^{sgd}(\theta_t^u)), \]

Eq. 7) \[ \theta_{t+1}^u = \theta_t^u - \eta \tilde{g}(\theta_t^u), \]

Eq. 14) \[ \theta_{t+1}^u = \theta_t^u - \eta \nabla_{\theta^u} L(\theta_t^u) + \eta \xi_t, \xi_t \sim \mathcal{N}(0, \Sigma^{sgd}(\theta_t^u)). \]

More SGD noise leads to better generalization.
Generalization Enhancement (GE)

The gradient follows a normal distribution as shown by the Central Limit Theorem

\[ \tilde{g}(\theta^u_t) \sim \mathcal{N}(\nabla_{\theta^u} L(\theta^u_t), \Sigma^{sgd}(\theta^u_t)), \]

Eq. 12

More SGD noise leads to better generalization.

\[ \theta^u_{t+1} = \theta^u_t - \eta \tilde{g}(\theta^u_t), \]

Eq. 7

\[ \theta^u_{t+1} = \theta^u_t - \eta \nabla_{\theta^u} L(\theta^u_t) + \eta \xi_t, \xi_t \sim \mathcal{N}(0, \Sigma^{sgd}(\theta^u_t)). \]

Eq. 14
Generalization Enhancement (GE)

The goal of GE is to replace lost SGD noise.

\[
\begin{align*}
\text{added noise} & \quad h(\theta^u_t) \sim \mathcal{N}(0, \Sigma^{\text{sgd}}(\theta^u_t)) \\
\text{Complete Equation} & \quad \theta^u_{t+1} = \theta^u_t - \eta (k^u_t \tilde{g}(\theta^u_t) + h(\theta^u_t))
\end{align*}
\]

*Eq. 16*

Eq. 14) \( \xi_t \sim \mathcal{N}(0, \Sigma^{\text{sgd}}(\theta^u_t)) \).  
Eq. 15) \( \xi'_t \sim \mathcal{N}(0, (k^u_t)^2 \cdot \Sigma^{\text{sgd}}(\theta^u_t)) \).  
Eq. 17) \( \xi''_t \sim \mathcal{N}(0, ((k^u_t)^2 + 1) \Sigma^{\text{sgd}}(\theta^u_t)) \).

*regular SGD*  
*With both OGM and GE*  
*With only OGM*
Generalization Enhancement (GE)

The goal of GE is to replace lost SGD noise.

\[ h(\theta^u_t) \sim \mathcal{N}(0, \Sigma^{sgd}(\theta^u_t)) \]

Eq. 16) \[ \theta^u_{t+1} = \theta^u_t - \eta (k^u_t \tilde{g}(\theta^u_t) + h(\theta^u_t)) \]

Complete Equation

Eq. 14) \[ \xi_t \sim \mathcal{N}(0, \Sigma^{sgd}(\theta^u_t)) \]

regular SGD

Eq. 15) \[ \xi'_t \sim \mathcal{N}(0, (k^u_t)^2 \cdot \Sigma^{sgd}(\theta^u_t)) \]

With only OGM

Eq. 17) \[ \xi''_t \sim \mathcal{N}(0, ((k^u_t)^2 + 1) \Sigma^{sgd}(\theta^u_t)) \]

With both OGM and GE
OGM-GE Algorithm

Methodology

Algorithm 1 Multimodal learning with OGM-GE strategy

**Input:** Training dataset $\mathcal{D} = \{(x_t^u, x_t^v, y_t)\}_{t=1,2,...,N}$, iteration number $T$, hyper-parameter $\alpha$, initialized modal-specific parameters $\theta^u$, $u \in \{a, v\}$.

for $t = 0, \ldots, T - 1$

Sample a fresh mini-batch $B_t$ from $\mathcal{D}$;

Feed-forward the batched data $B_t$ to the model;

Calculate $\rho^u$ using Equation 8 and 9;

Calculate $k_t^u$ using Equation 10;

Calculate gradient $\tilde{g}(\theta_t^u)$ using back-propagation;

Sample $h(\theta_t^u)$ based on covariance of gradient $\tilde{g}(\theta_t^u)$;

Update using $\theta_{t+1}^u = \theta_t^u - \eta(k_t^u \tilde{g}(\theta_t^u) + h(\theta_t^u))$.

end for
Experimental Setup
Datasets
Multi-modal categorization
1. CREMA-D
2. Kinetics-Sounds
3. VGGSound
Audio-Visual Localization
4. AVE

Experimental Setup
Experimental Settings

Audio Encoder
• Transformed to spectrogram
• ResNet18-based
• Input channels set to 1

Visual Encoder
• ResNet18-based
• 3 frames per video
• Temporal Pooling
Results
How does OGM-GE compare to conventional fusion methods?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CREMA-D</th>
<th></th>
<th>VGGSound</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Acc</td>
<td>mAP</td>
<td>Acc</td>
<td>mAP</td>
</tr>
<tr>
<td>Audio-only</td>
<td>52.5</td>
<td>54.2</td>
<td>44.3</td>
<td>48.4</td>
</tr>
<tr>
<td>Visual-only</td>
<td>41.9</td>
<td>43.0</td>
<td>31.0</td>
<td>34.3</td>
</tr>
<tr>
<td>Baseline</td>
<td>50.8</td>
<td>52.6</td>
<td>48.4</td>
<td>51.7</td>
</tr>
<tr>
<td>Concatenation</td>
<td>51.7</td>
<td>53.5</td>
<td>49.1</td>
<td>52.5</td>
</tr>
<tr>
<td>Summation</td>
<td>51.5</td>
<td>53.5</td>
<td>49.1</td>
<td>52.4</td>
</tr>
<tr>
<td>FiLM [32]</td>
<td>50.6</td>
<td>52.1</td>
<td>48.5</td>
<td>51.6</td>
</tr>
<tr>
<td>Baseline†</td>
<td>54.4</td>
<td>56.2</td>
<td>50.1</td>
<td>53.5</td>
</tr>
<tr>
<td>Concatenation†</td>
<td>61.9</td>
<td>63.9</td>
<td>50.6</td>
<td>53.9</td>
</tr>
<tr>
<td>Summation†</td>
<td>62.2</td>
<td>64.3</td>
<td>50.4</td>
<td>53.6</td>
</tr>
<tr>
<td>FiLM†</td>
<td>55.6</td>
<td>57.4</td>
<td>50.0</td>
<td>52.9</td>
</tr>
</tbody>
</table>

OGM-GE consistently improves performance of baseline methods.

† indicates that OGM-GE was applied
How does OGM-GE compare to other modulation strategies?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CREMA-D</th>
<th>KS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concatenation</td>
<td>51.7</td>
<td>59.8</td>
</tr>
<tr>
<td>Modality-Drop [9] (audio)</td>
<td>54.4</td>
<td>60.3</td>
</tr>
<tr>
<td>Modality-Drop [9] (visual)</td>
<td>53.3</td>
<td>61.3</td>
</tr>
<tr>
<td>Grad-Blending [39]</td>
<td>56.8</td>
<td>62.2</td>
</tr>
<tr>
<td>OGM</td>
<td>59.0</td>
<td>61.1</td>
</tr>
<tr>
<td>OGM-GE</td>
<td>61.9</td>
<td>62.3</td>
</tr>
</tbody>
</table>

All methods make progress, but OGM-GE achieves the highest performance.
Can OGM-GE be **combined** with existing methods?

All multimodal approaches evaluated show improvements with OGM-GE

<table>
<thead>
<tr>
<th>Dataset</th>
<th>KS Acc</th>
<th>VGGSound Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSN-AV [38]</td>
<td>58.6</td>
<td>49.0</td>
</tr>
<tr>
<td>TSM-AV [26]</td>
<td>60.3</td>
<td>48.8</td>
</tr>
<tr>
<td>TBN [24]</td>
<td>60.8</td>
<td>49.4</td>
</tr>
<tr>
<td>PSP [46]</td>
<td>59.7</td>
<td>49.2</td>
</tr>
<tr>
<td>TSN-AV†</td>
<td>59.1</td>
<td>49.6</td>
</tr>
<tr>
<td>TSM-AV†</td>
<td>62.4</td>
<td>49.6</td>
</tr>
<tr>
<td>TBN†</td>
<td><strong>63.1</strong></td>
<td><strong>50.4</strong></td>
</tr>
<tr>
<td>PSP†</td>
<td>60.4</td>
<td>49.5</td>
</tr>
</tbody>
</table>

OGM-GE is not limited to disconnected encoders
PSP is an example using co-attention.

Results for the CREMA-D dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-vector [15]</td>
<td>53.6</td>
</tr>
<tr>
<td>X-vector [30]</td>
<td>55.6</td>
</tr>
<tr>
<td>MWTS[12]</td>
<td>54.1</td>
</tr>
<tr>
<td>I-vector†</td>
<td>55.3</td>
</tr>
<tr>
<td>X-vector†</td>
<td>57.1</td>
</tr>
<tr>
<td>MWTS[†]</td>
<td><strong>58.0</strong></td>
</tr>
</tbody>
</table>

† indicates that OGM-GE was applied
Can OGM-GE be applied to other tasks?

OGM-GE can also work on audio-visual event localization (AVE).

<table>
<thead>
<tr>
<th>Audio-visual Event Localization</th>
<th>w/o OGM-GE</th>
<th>w/ OGM-GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVGA [36]</td>
<td>72.0</td>
<td>72.8</td>
</tr>
<tr>
<td>PSP [46]</td>
<td>76.2</td>
<td>76.9</td>
</tr>
</tbody>
</table>
Ablation Study

OGM-GE still improves performance when used with an Adam optimizer.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CREMA-D</th>
<th>KS</th>
<th>VGGSound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Acc</td>
<td>Acc</td>
<td>Acc</td>
</tr>
<tr>
<td>SGD</td>
<td>51.7</td>
<td>59.8</td>
<td>49.1</td>
</tr>
<tr>
<td>SGD†</td>
<td>61.9</td>
<td>63.1</td>
<td>50.6</td>
</tr>
<tr>
<td>Adam</td>
<td>49.7</td>
<td>57.4</td>
<td>47.3</td>
</tr>
<tr>
<td>Adam†</td>
<td>54.6</td>
<td>58.9</td>
<td>48.2</td>
</tr>
</tbody>
</table>

Evaluation of learning rates and batch sizes.

<table>
<thead>
<tr>
<th>Settings</th>
<th>CREMA-D</th>
<th>VGGSound</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b=64, lr=1e-4)</td>
<td>50.4</td>
<td>48.3</td>
</tr>
<tr>
<td>(b=64, lr=5e-4)</td>
<td>51.0</td>
<td>48.7</td>
</tr>
<tr>
<td>(b=64, lr=1e-3)</td>
<td>51.8</td>
<td>49.1</td>
</tr>
<tr>
<td>(b=64, lr=1e-3)†</td>
<td>51.8</td>
<td>49.1</td>
</tr>
<tr>
<td>(b=128, lr=1e-3)</td>
<td>50.2</td>
<td>48.8</td>
</tr>
<tr>
<td>(b=256, lr=1e-3)</td>
<td>48.6</td>
<td>47.7</td>
</tr>
<tr>
<td>(b=64, lr=1e-3) w/ GE</td>
<td>60.2</td>
<td>50.3</td>
</tr>
</tbody>
</table>
Strengths & Weaknesses
Strengths

• No limited by modality
  - Applicable to text, full video, etc.
• The authors address the potential issue of interconnected encoders by evaluating PSP.
• Limited computational requirements, and fairly easy to implement.
• Consistently improves baseline results.

Evaluation of VGGSound dataset
Weaknesses

- Algorithm seems to require a categorical output
  - Event localization is implemented as “fine-grained classification”.

- Minimum penalty is greater than 0.
  - May be a problem for near-equal settings.

- No method prescribed for optimizing degree of modulation $\alpha$.
  - Ranges from 0.1 to 0.8

- Individual components still fall short of unimodal training
Discussion
Discussion Questions

1. Are the gains produced by adding Gaussian noise dependent on the modulation? (i.e. would running only GE improve the results)

2. How would you optimize the degree of modulation $\alpha$, or make the algorithm choose it dynamically?

3. The authors claim the method is not limited to models with separated encoders, what architectures might cause OGM-GE to fail?
References

https://visualqa.org/