Spatial Sparse CNNs from Masks

Shuyue Jia

Challenges & Problems:

- **Region of Interest**
  - Attention Mechanism
  - Image regions are not equally important
- **Spatial sparsity**
  - Traditional (Dense) Convolutions → high computational cost
  - Binary masks → Sparse Region of Interest
- **Practical Speed-up**
  - Many literature: theoretical complexity
  - Slow inference speed

Related Work (Conditional Execution / NN Gating):

- **Layer-based methods:** Certain network layers or blocks
  - Adaptive Computation Time ← Stop learning (halting score)
- **Channel-based methods:** Prune channels dynamically
  - Advanced features are only needed for a subset of the images
- **Spatial methods:**
  - Glimpse/Cascades → Region of Interest **but** Lose features
  - Spatially Adaptive Computation Time (SACT) ← features refinement
  - SBNet (Two stage): Mask → Tiles
    - Masks → Attention Mechanism (Weights are binary)

Methods (training):

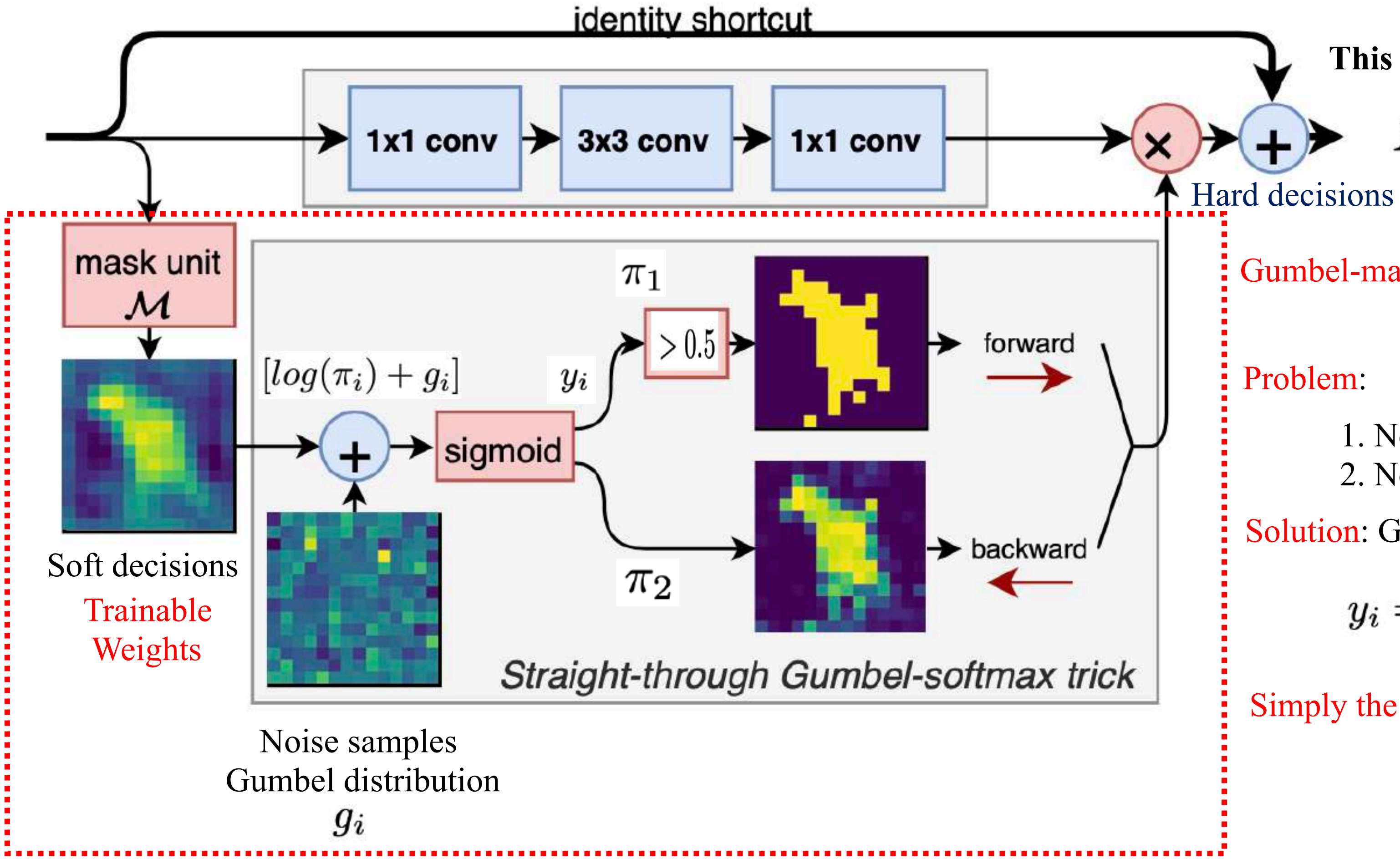
Traditional Method: direct CNNs (ResNet) ← Feature Extraction

$$X_{b+1} = r(\mathcal{F}(X_b) + X_b)$$

$X_b \in (w_{b+1} \times h_{b+1})$

Class

π_i



This work: Conditional Spatial Sparse CNNs

$$X_{b+1} = r(\mathcal{F}(X_b) \circ G_b + X_b)$$

Gumbel-max Sampling: category distribution

$$\arg\max_i \, [log(\pi_i) + g_i]$$

Problem:

1. Not a good probability distribution
2. Not continuous, differentiable

Solution: Gumbel-Softmax Sampling

$$y_i = \frac{\exp((log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((log(\pi_j) + g_j)/\tau)}$$

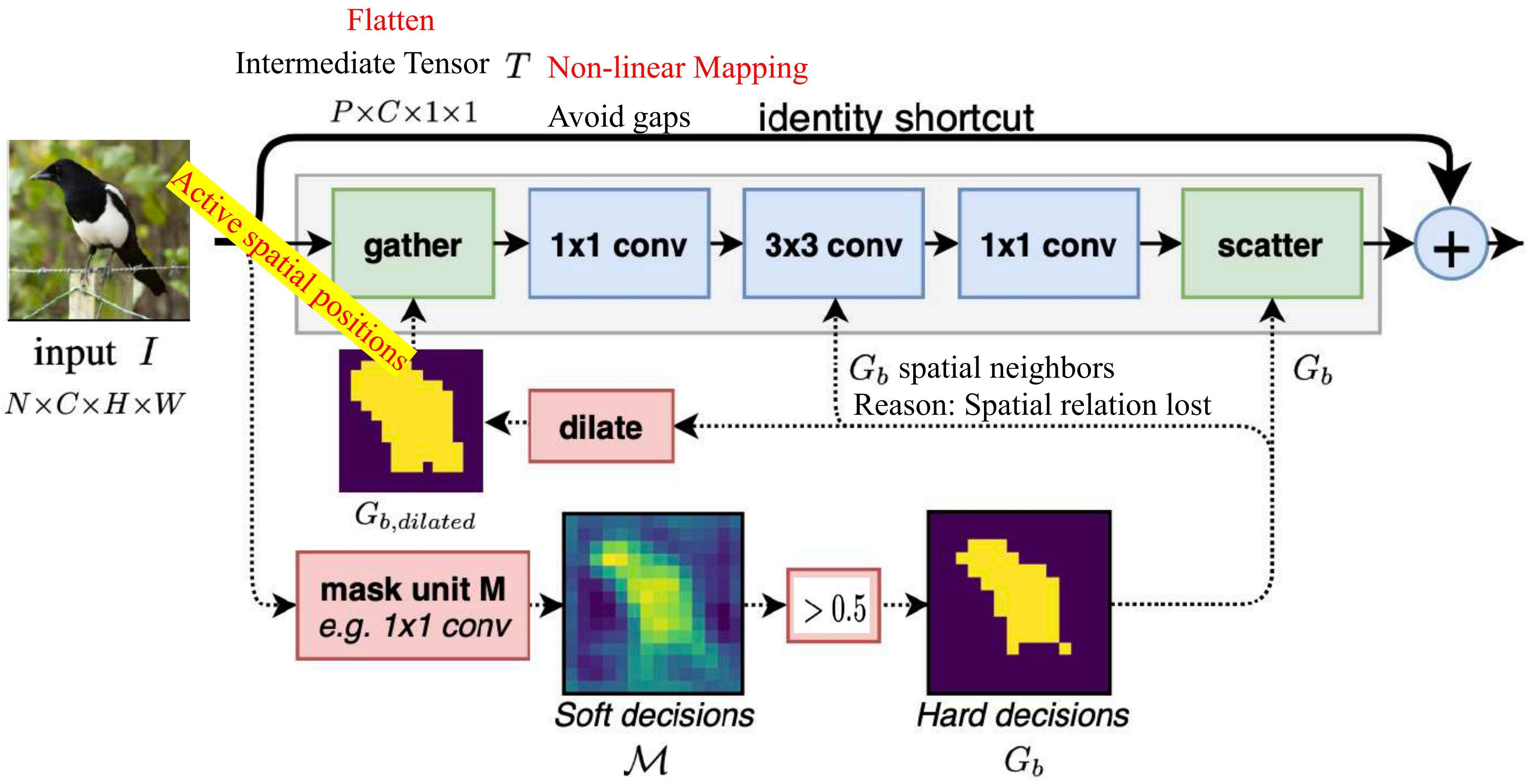
Simply the equation:

$$y_1 = \sigma\left(\frac{m + g_1 - g_2}{\tau}\right) > 0.5$$

$$G_b = \mathcal{G}(\mathcal{M}(X_b)) \in \{0,1\}^{w_{b+1} \times h_{b+1}}$$

Goal: **study the spatial execution masks for an image**

Methods (Inference): Gather-scatter Strategy



Loss Function: sparsity loss criterion

MobileNetV2 : $\mathbb{F}_b = H \cdot W \cdot \left(9C_{b,e} + 2C_bC_{b,e}\right)$

This Work: $\mathbb{F}_{b,sp} = N_{b,dilated}C_bC_{b,e} + N_b\left(9C_{b,e} + C_{b,e}C_b\right)$ $\quad N_b = \sum G_b$

Floating point Operations Loss

$\theta \in [0,1]$

$$\mathcal{L}_{sp,net} = \left(\frac{\sum_b^B \mathbb{F}_{b,sp}}{\sum_b^B \mathbb{F}_b} - \theta\right)^2$$

$$\mathcal{L}_{sp,low} = \frac{1}{B}\sum_b^B \max(0, p \cdot \theta - \frac{\mathbb{F}_{b,sp}}{\mathbb{F}_b})^2$$

$$\mathcal{L}_{sp,up} = \frac{1}{B}\sum_b^B \max(0, \frac{\mathbb{F}_{b,sp}}{\mathbb{F}_b} - (1 - p(1-\theta)))^2 \qquad p \in [0,1]$$

$$\mathcal{L} = \mathcal{L}_{task} + \boxed{\alpha(\mathcal{L}_{sp,net} + \mathcal{L}_{sp,lower} + \mathcal{L}_{sp,upper})}$$

Limitations and Improvements

- **Limitations:**
  - **Applications:**
    - Smaller Objects ← Gather Operation (Flatten)
    - Multiple Objects
    - Background Clutter
    - etc.
  - **Algorithms:**
    - Features cannot be fully extracted ← Region of Proposal (Musk)
- **Potential Improvements:**
  - Transformers ← Attention Mechanism (Reference: "End-to-End Object Detection with Transformers")
  - Fine-grained features extracted methods
  - If 3D Convolution: Factored Convolution $O(N^3) \rightarrow O(N^2+N)$ Speed up

3D Human Pose Estimation by

Mixing **2D Image** and **3D Depth** **Triplets Heatmaps**

Challenges & Problems:

- **Lack of Information (Features)**
  - Single Image ← inherent ambiguities
  - Attention Mechanism
- **Hard to trade-off between Efficiency and Effectiveness**
  - Representation efficiency
  - Learning effectiveness
- **Lack of Training Data**
  - Manual annotation → “In the wild” Images
  - 3D Annotations

Related Work (3D pose estimation based on CNNs):

- **Direct Encoder-Decoder**
  - Single stage
  - End-to-end
- **Transition with 2D Joints**
  - Two stages
  - 2D image → 2D joint locations → 3D space (3D joint locations)
- **3D-Aware Intermediate States**
  - Two stages
  - 2D image → **3D-aware states** → 3D joint locations
  - Volumetric Representation

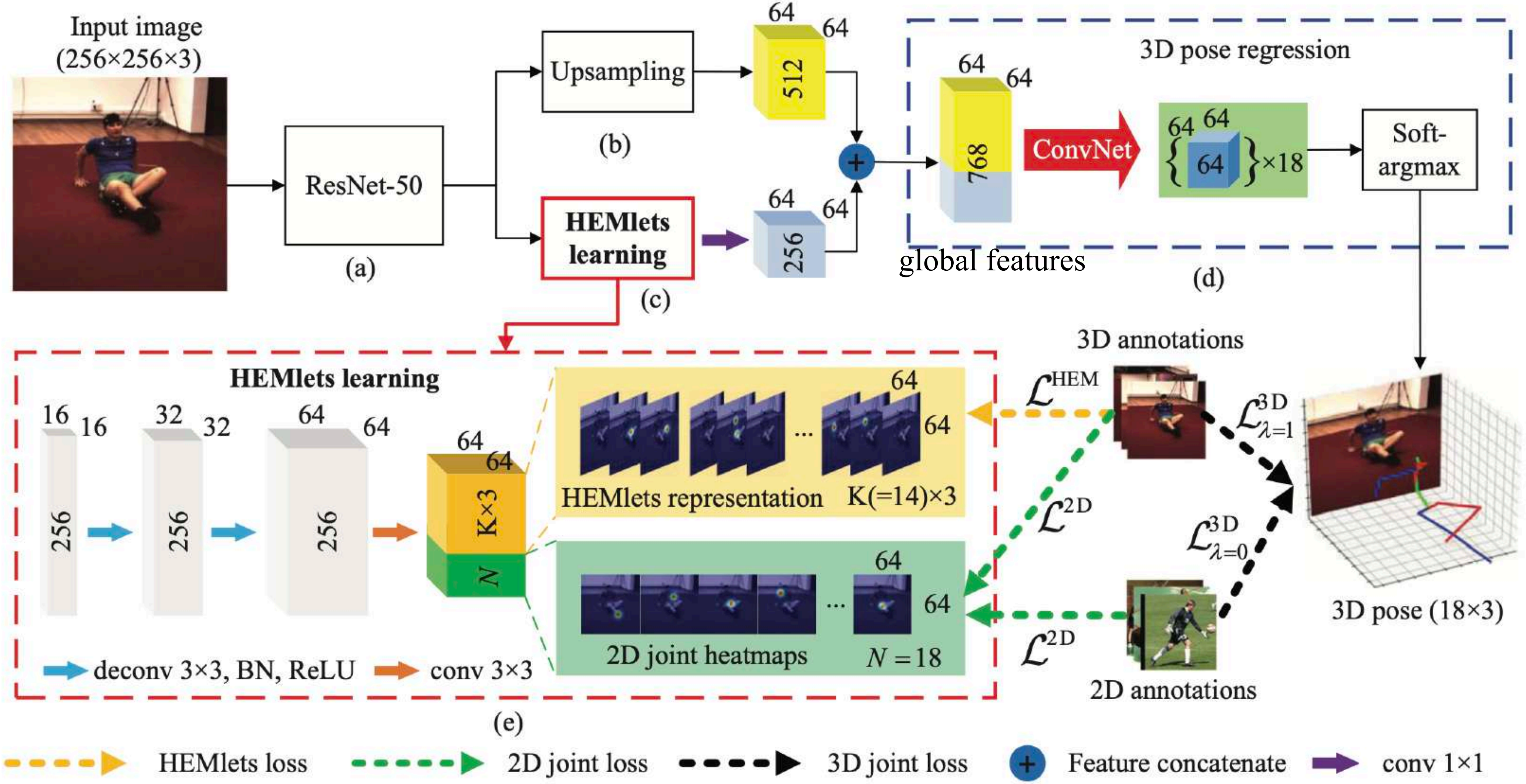
* **Helpful**: **Relative depth information** (This work: Part-Centric Heatmap Triplets) → Promote Performance

Related Work (3D human body reconstruction based on CNNs):

*** Parametric human body space, e.g., SMPL**

- **Two-stage Framework**
  - 2D image → 2D joint locations → SMPL
  - Depth ambiguity → Local minimum
- **One-stage Framework**
  - 2D image → SMPL
  - Lack of 3D model annotations
- **Intermediate States**
  - Two stages
  - 2D image → **2D Intermediate states** → SMPL
- **Voxel, Mesh, UV-maps**

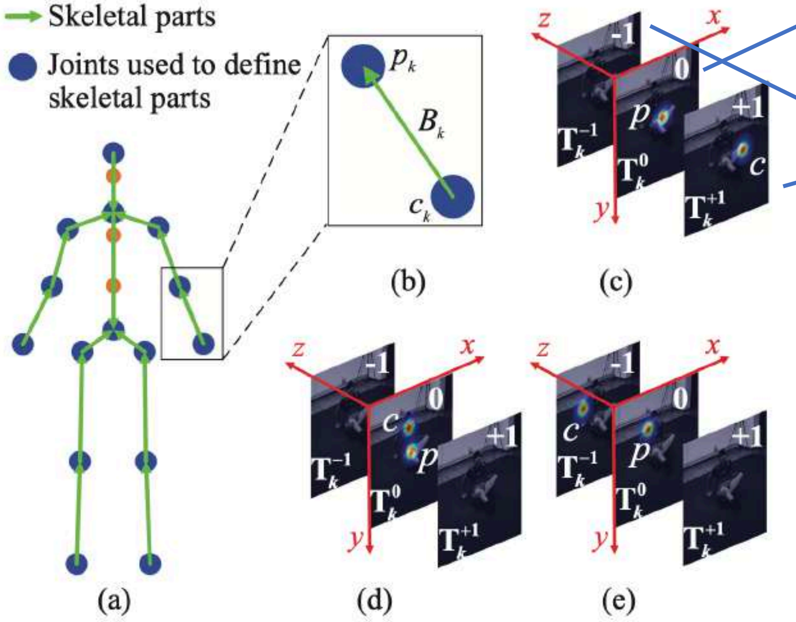
Methods:



Input image
(256×256×3)
ResNet-50
(a)
Upsampling
(b)
HEMlets learning
(c)
512
256
64
3D pose regression
768
ConvNet
64
×18
Soft-argmax
global features
(d)
3D annotations
3D pose (18×3)
2D annotations
HEMlets learning
16
32
256
K×3
N
HEMlets representation
K(=14)×3
2D joint heatmaps
N = 18
deconv 3×3, BN, ReLU
conv 3×3
(e)
HEMlets loss
2D joint loss
3D joint loss
Feature concatenate
conv 1×1

Methods: Intermediate representation of the 3D-aware relationship

- **2D Image (coordinates)**
- **Relative depth information ← Part-Centric Heatmap Triplets**



p: parent joint

c: child joint

Part-centric heatmap triplets $\{\mathbf{T}_k^{-1}, \mathbf{T}_k^{0}, \mathbf{T}_k^{+1}\}$

$$\mathbf{T}_k = \mathrm{Stack}[\mathbf{T}_k^{-1}, \mathbf{T}_k^{0}, \mathbf{T}_k^{+1}]$$

$$P(x_p, y_p, x_c, y_c, r(z_p, z_c))$$

1. Pairwise joints' co-location likelihoods
2. Depth relations → learn geometric constraints

Relative Depth Ordering

$$r(z_p, z_c) = \begin{cases} 1 & z_p - z_c > \epsilon \\ 0 & |z_p - z_c| < \epsilon \\ -1 & z_p - z_c < -\epsilon \end{cases}$$

ϵ : Relative depth difference

Loss Function:

HEMlets loss: $\mathcal{L}^{\mathrm{HEM}} = ||(\mathbf{T}^{\mathrm{gt}} - \hat{\mathbf{T}}) \odot \mathbf{\Lambda}||_2^2$

Auxiliary 2D joint loss: $\mathcal{L}^{\mathrm{2D}} = \sum_{n=1}^{N} ||\mathbf{H}_n^{\mathrm{gt}} - \hat{\mathbf{H}}_n||_2^2$

Soft-argmax 3D joint loss: $[\hat{x}_n, \hat{y}_n, \hat{z}_n] = \int_{\mathbf{v}} \mathbf{v} \cdot \mathrm{Softmax}(\mathbf{F}_n)$

3D joints Regression loss: $\mathcal{L}_\lambda^{\mathrm{3D}} = \sum_{n=1}^{N} (\left|x_n^{\mathrm{gt}} - \hat{x}_n\right| + \left|y_n^{\mathrm{gt}} - \hat{y}_n\right| + \lambda \left|z_n^{\mathrm{gt}} - \hat{z}_n\right|)$

Training Loss: $\mathcal{L}^{\mathrm{int}} = \mathcal{L}^{\mathrm{HEM}} + \mathcal{L}^{\mathrm{2D}}$

$$\mathcal{L}^{\mathrm{tot}} = \alpha * \mathcal{L}^{\mathrm{int}} + \mathcal{L}_\lambda^{\mathrm{3D}}$$

Limitations and Improvements

- **Limitations:**
  - **Algorithms:**
    - Heatmaps? ← Region of Interest
    - Not efficient ← 2D joint annotations and 3D joint annotations
    - Hard to transfer to other objects ← Too many annotations
- **Potential Improvements:**
  - Combine the last paper: Spatial Sparse CNNs from Masks → Heatmaps
  - Depth Information should be considered