ECCV'20 ONLINE

23-28 AUGUST 2020

16TH EUROPEAN CONFERENCE ON

COMPUTER VISION

WWW.ECCV2020.EU

Energy-Based Models for Deep Probabilistic Regression

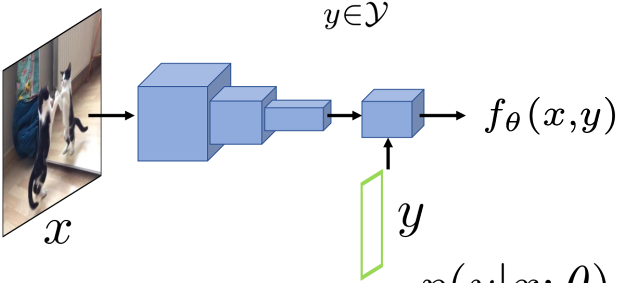
Fredrik K. Gustafsson[1] Martin Danelljan[2] Goutam Bhat[2] Thomas B. Schön[1]

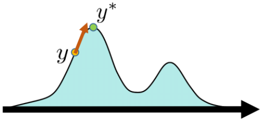
[1]Department of Information Technology, Uppsala University, Sweden
[2]Computer Vision Lab, ETH Zürich, Switzerland

UPPSALA UNIVERSITET

We achieve state-of-the-art regression performance on four diverse tasks by employing **energy-based models (EBMs)** within a probabilistic formulation. Our proposed approach is conceptually simple and straightforward to both implement and train.

$$y^* = \arg\max_{y \in \mathcal{Y}} p(y|x;\theta) = \arg\max_{y \in \mathcal{Y}} f_\theta(x,y)$$





$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{\int e^{f_\theta(x,\tilde{y})} d\tilde{y}}$$

Outline

UPPSALA UNIVERSITET

Supervised Regression: learn to predict a continuous target value $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

Supervised Regression: learn to predict a continuous target value $y^\star \in \mathcal{Y} = \mathbb{R}^K$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set $\mathcal{D}$ of i.i.d. input-target examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$.

Deep Neural Network (DNN): a function $f_\theta : \mathcal{U} \to \mathcal{O}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input $u \in \mathcal{U}$ to an output $f_\theta(u) \in \mathcal{O}$.

Direct Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

1.1 Direct Regression

Direct Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\operatorname{argmin}}\, J(\theta').$$

UPPSALA UNIVERSITET

Direct Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\operatorname{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Direct Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\text{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Minimizing $J(\theta)$ then corresponds to minimizing the *negative log-likelihood* $\sum_{i=1}^{N} -\log p(y_i | x_i; \theta)$ for a specific model $p(y | x; \theta)$ of the conditional target density.

Direct Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{Y}$ to directly predict the target, $y^\star = f_\theta(x^\star)$.

The DNN model parameters θ are learned by minimizing a loss function $\ell(f_\theta(x_i), y_i)$, penalizing discrepancy between the prediction $f_\theta(x_i)$ and the ground truth y_i:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(x_i), y_i), \quad \theta = \underset{\theta'}{\mathrm{argmin}}\, J(\theta').$$

The most common choices for ℓ are the L^2 loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, and the L^1 loss.

Minimizing $J(\theta)$ then corresponds to minimizing the *negative log-likelihood* $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$ for a specific model $p(y|x;\theta)$ of the conditional target density.

For example, the L^2 loss corresponds to a fixed-variance Gaussian model (1D case):

$$p(y|x;\theta) = \mathcal{N}(y; f_\theta(x), \sigma^2).$$

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

UPPSALA UNIVERSITET

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \rightarrow \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

For example, a general 1D Gaussian model can be realized as:

$$p(y|x;\theta) = \mathcal{N}\big(y;\mu_\theta(x),\, \sigma^2_\theta(x)\big), \quad f_\theta(x) = [\,\mu_\theta(x) \quad \log\sigma^2_\theta(x)\,]^\mathsf{T} \in \mathbb{R}^2.$$

Why not explicitly employ this probabilistic perspective and try to create more *flexible* models $p(y|x;\theta)$ of the conditional target density $p(y|x)$?

Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \to \mathcal{O}$ to predict the parameters ϕ of a certain family of probability distributions $p(y;\phi)$, then model $p(y|x)$ with:

$$p(y|x;\theta) = p(y;\phi(x)), \quad \phi(x) = f_\theta(x).$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

For example, a general 1D Gaussian model can be realized as:

$$p(y|x;\theta) = \mathcal{N}\big(y;\mu_\theta(x),\sigma_\theta^2(x)\big), \quad f_\theta(x) = [\,\mu_\theta(x) \quad \log\sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2.$$

The negative log-likelihood $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$ then corresponds to the loss:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{(y_i - \mu_\theta(x_i))^2}{\sigma_\theta^2(x_i)} + \log\sigma_\theta^2(x_i).$$

1.3 Confidence-Based Regression

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

1.3 Confidence-Based Regression

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-Based Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity w.r.t. y to predict the target:

$$y^\star = \operatorname*{argmax}_y f_\theta(x^\star, y)$$

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-Based Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity w.r.t. y to predict the target:

$$y^\star = \operatorname*{argmax}_y f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-Based Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity w.r.t. y to predict the target:

$$y^\star = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

Commonly employed for image-coordinate regression, e.g. human pose estimation [15], where the DNN predicts a 2D confidence heatmap over image-coordinates y.

1.3 Confidence-Based Regression

The quest for improved regression accuracy has also led to the development of more specialized methods, achieving state-of-the-art performance within computer vision.

Confidence-Based Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar confidence value $f_\theta(x, y)$, and maximize this quantity w.r.t. y to predict the target:

$$y^\star = \operatorname*{argmax}_y f_\theta(x^\star, y)$$

The DNN model parameters θ are learned by generating *pseudo* ground truth confidence values $c(x_i, y_i, y)$, and minimizing a loss function $\ell\big(f_\theta(x_i, y), c(x_i, y_i, y)\big)$.

Commonly employed for image-coordinate regression, e.g. human pose estimation [15], where the DNN predicts a 2D confidence heatmap over image-coordinates y. Recently, the approach was also employed by IoU-Net [7] for bounding box regression in object detection, which in turn was utilized by the ATOM [4] visual tracker.

Outline

UPPSALA UNIVERSITET

2. Proposed Regression Method

UPPSALA UNIVERSITET

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation.

2. Proposed Regression Method

UPPSALA UNIVERSITET

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

In this work, we aim to combine the benefits of these two approaches.

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

In this work, we aim to combine the benefits of these two approaches.

EBMs for Deep Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with the EBM:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

2. Proposed Regression Method

While **confidence-based regression** methods have demonstrated impressive results, they require important task-dependent design choices (e.g. how to generate the pseudo ground truth labels) and usually lack a clear probabilistic interpretation. In contrast, the framework of **probabilistic regression** is straightforward and generally applicable, but can usually not compete in terms of regression accuracy.

In this work, we aim to combine the benefits of these two approaches.

EBMs for Deep Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with the EBM:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

UPPSALA UNIVERSITET

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Training thus requires the evaluation of $Z(x,\theta)$, we employ importance sampling:

UPPSALA UNIVERSITET

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

Training thus requires the evaluation of $Z(x,\theta)$, we employ importance sampling:

$$\begin{aligned} -\log p(y_i|x_i;\theta) &= \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i,y_i) \\ &= \log\left(\int \frac{e^{f_\theta(x_i,y)}}{q(y)} q(y) dy\right) - f_\theta(x_i,y_i) \\ &\approx \log\left(\frac{1}{M}\sum_{k=1}^{M} \frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y). \end{aligned}$$

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M} \frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M} \frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

We use a proposal distribution $q(y) = q(y|y_i) = \frac{1}{L}\sum_{l=1}^{L} \mathcal{N}(y; y_i, \sigma_l^2)$ that depends on the ground truth target y_i.

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

The DNN model parameters θ are learned by minimizing $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$.

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M}\frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)})}\right) - f_\theta(x_i,y_i), \quad y^{(k)} \sim q(y).$$

We use a proposal distribution $q(y) = q(y|y_i) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(y; y_i, \sigma_l^2)$ that depends on the ground truth target y_i. The final minimization objective $J(\theta)$ is thus given by:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i,y^{(i,m)})}}{q(y^{(i,m)}|y_i)}\right) - f_\theta(x_i,y_i), \quad \{y^{(i,m)}\}_{m=1}^{M} \sim q(y|y_i).$$

Our EBM $p(y|x;\theta) = e^{f_\theta(x,y)}/Z(x,\theta)$ is highly flexible and can learn complex target densities directly from data, including multi-modal and asymmetric densities.

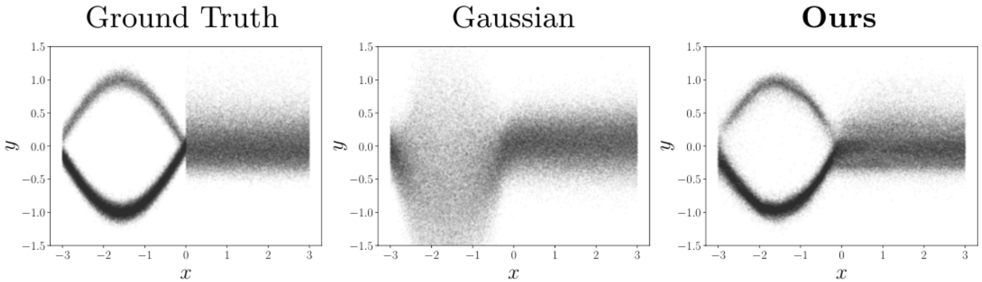


Figure 2: An illustrative 1D regression problem. The training data $\{(x_i, y_i)\}_{i=1}^{2000}$ is generated by the ground truth conditional target density $p(y|x)$.

UPPSALA UNIVERSITET

EBMs for Deep Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with the EBM:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

EBMs for Deep Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with the EBM:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Given an input $x^\star$ at test time, we predict the target $y^\star$ by maximizing $p(y|x^\star;\theta)$:

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

EBMs for Deep Probabilistic Regression: train a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to predict a scalar value $f_\theta(x, y)$, then model $p(y|x)$ with the EBM:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

Given an input $x^\star$ at test time, we predict the target $y^\star$ by maximizing $p(y|x^\star;\theta)$:

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

By designing the DNN f_θ to be differentiable w.r.t. targets y, the gradient $\nabla_y f_\theta(x^\star, y)$ can be efficiently evaluated using auto-differentiation. We can thus perform *gradient ascent* to find a local maximum of $f_\theta(x^\star, y)$, starting from an initial estimate $\hat{y}$.

UPPSALA UNIVERSITET

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

EBMs for Deep Probabilistic Regression:

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \quad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}.$$

$$y^\star = \underset{y}{\operatorname{argmax}}\, p(y|x^\star;\theta) = \underset{y}{\operatorname{argmax}}\, f_\theta(x^\star, y).$$

Algorithm S1 Prediction via gradient-based refinement.

Input: $x^\star$, $\hat{y}$, T, λ, η.

1: $y \leftarrow \hat{y}$.
2: **for** $t = 1, \dots, T$ **do**
3:     `PrevValue` $\leftarrow f_\theta(x^\star, y)$.
4:     $\tilde{y} \leftarrow y + \lambda \nabla_y f_\theta(x^\star, y)$.
5:     `NewValue` $\leftarrow f_\theta(x^\star, \tilde{y})$.
6:     **if** `NewValue` $>$ `PrevValue` **then**
7:         $y \leftarrow \tilde{y}$.
8:     **else**
9:         $\lambda \leftarrow \eta\lambda$.
10: **Return** y.

Outline

We evaluate our proposed approach on four diverse computer vision regression tasks: **object detection**, **visual tracking**, **age estimation** and **head-pose estimation**.

We evaluate our proposed approach on four diverse computer vision regression tasks: **object detection**, **visual tracking**, **age estimation** and **head-pose estimation**.

Our approach significantly outperforms the state-of-the-art confidence-based IoU-Net [7] method for bounding box regression in *direct comparisons*, both when applied for object detection on COCO [10], and in the ATOM [4] visual tracker.

We evaluate our proposed approach on four diverse computer vision regression tasks: **object detection**, **visual tracking**, **age estimation** and **head-pose estimation**.

Our approach significantly outperforms the state-of-the-art confidence-based IoU-Net [7] method for bounding box regression in *direct comparisons*, both when applied for object detection on COCO [10], and in the ATOM [4] visual tracker.

In contrast to confidence-based methods, our approach is shown to also be *directly applicable* to more general tasks such as age and head-pose estimation.

UPPSALA UNIVERSITET

Object Detection: when applied to refine the Faster-RCNN detections on COCO, our approach both significantly improves the original detections and outperforms IoU-Net.

Formulation Approach	Direct Faster-RCNN [9]	Gaussian	Gaussian Mixt. 2	Gaussian Mixt. 4	Gaussian Mixt. 8	Gaussian cVAE	Laplace	Confidence IoU-Net [7]	Confidence IoU-Net*	**Ours**
AP (%)	37.2	36.7	37.1	37.0	36.8	37.2	37.1	38.3	38.2	**39.4**
AP_{50}(%)	**59.2**	58.7	59.1	59.1	59.1	**59.2**	59.1	58.3	58.4	58.6
AP_{75}(%)	40.3	39.6	40.0	39.9	39.7	40.0	40.2	41.4	41.4	**42.1**
FPS	**12.2**	**12.2**	**12.2**	12.1	12.1	9.6	**12.2**	5.3	5.3	5.3

Visual Tracking: when applied to refine the initial estimate provided by the classifier in ATOM, our approach significantly outperforms the original IoU-Net-based method. Our approach also outperforms other state-of-the-art trackers.

Dataset	Metric	ECO [5]	SiamFC [1]	MDNet [13]	UPDT [2]	DaSiamRPN [17]	SiamRPN++ [8]	ATOM [4]	ATOM*	**Ours**
TrackingNet [12]	Precision (%)	49.2	53.3	56.5	55.7	59.1	69.4	64.8	66.6	**69.7**
	Norm. Prec. (%)	61.8	66.6	70.5	70.2	73.3	80.0	77.1	78.4	**80.1**
	Success (%)	55.4	57.1	60.6	61.1	63.8	73.3	70.3	72.0	**74.5**
UAV123 [11]	$OP_{0.50}$ (%)	64.0	-	-	66.8	73.6	$75^{\dagger}$	78.9	79.0	**80.8**
	$OP_{0.75}$ (%)	32.8	-	-	32.9	41.1	$56^{\dagger}$	55.7	56.5	**60.2**
	AUC (%)	53.7	-	52.8	55.0	58.4	61.3	65.0	64.9	**67.2**

UPPSALA UNIVERSITET

Age Estimation: refinement using our proposed method consistently improves MAE (lower is better) for the age predictions outputted by a number of baselines.

+Refine	Niu et al. [14]	Cao et al. [3]	Direct	Gaussian	Laplace	Softmax (CE, L^2)	Softmax (CE, L^2, Var)
	5.74 $\pm$ 0.05	5.47 $\pm$ 0.01	4.81 $\pm$ 0.02	4.79 $\pm$ 0.06	4.85 $\pm$ 0.04	4.78 $\pm$ 0.05	4.81 $\pm$ 0.03
✓	-	-	**4.65** $\pm$ 0.02	4.66 $\pm$ 0.04	4.81 $\pm$ 0.04	**4.65** $\pm$ 0.04	4.69 $\pm$ 0.03

Head-Pose Estimation: refinement using our method consistently improves the average MAE for yaw, pitch and roll for the predicted pose outputted by our baselines.

+Refine	Gu et al. [6]	Yang et al. [16]	Direct	Gaussian	Laplace	Softmax (CE, L^2)	Softmax (CE, L^2, Var)
	3.66	3.60	3.09 $\pm$ 0.07	3.12 $\pm$ 0.08	3.21 $\pm$ 0.06	3.04 $\pm$ 0.08	3.15 $\pm$ 0.07
✓	-	-	3.07 $\pm$ 0.07	3.11 $\pm$ 0.07	3.19 $\pm$ 0.06	**3.01** $\pm$ 0.07	3.11 $\pm$ 0.06

[1] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision (ECCV) Workshops*, 2016.

[2] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg. Unveiling the power of deep tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–498, 2018.

[3] W. Cao, V. Mirjalili, and S. Raschka. Rank-consistent ordinal regression for neural networks. *arXiv preprint arXiv:1901.07884*, 2019.

[4] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4660–4669, 2019.

[5] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. ECO: Efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6638–6646, 2017.

[6] J. Gu, X. Yang, S. De Mello, and J. Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1548–1557, 2017.

[7] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[8] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4282–4291, 2019.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[11] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for UAV tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 445–461, 2016.

[12] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018.

[13] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, 2016.

[14] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928, 2016.

[15] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[16] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang. FSA-Net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1087–1096, 2019.

[17] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.

Contact

UPPSALA UNIVERSITET

Fredrik K. Gustafsson, Uppsala University

fredrik.gustafsson@it.uu.se

www.fregu856.com