# Energy-Based Models for Deep Probabilistic Regression

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

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

**Abstract.** While deep learning-based classification is generally tackled using standardized approaches, a wide variety of techniques are employed for regression. In computer vision, one particularly popular such technique is that of confidence-based regression, which entails predicting a confidence value for each input-target pair $(x, y)$. While this approach has demonstrated impressive results, it requires important task-dependent design choices, and the predicted confidences lack a natural probabilistic meaning. We address these issues by proposing a general and conceptually simple regression method with a clear probabilistic interpretation. In our proposed approach, we create an energy-based model of the conditional target density $p(y|x)$, using a deep neural network to predict the un-normalized density from $(x, y)$. This model of $p(y|x)$ is trained by directly minimizing the associated negative log-likelihood, approximated using Monte Carlo sampling. We perform comprehensive experiments on four computer vision regression tasks. Our approach outperforms direct regression, as well as other probabilistic and confidence-based methods. Notably, our model achieves a 2.2% AP improvement over Faster-RCNN for object detection on the COCO dataset, and sets a new state-of-the-art on visual tracking when applied for bounding box estimation. In contrast to confidence-based methods, our approach is also shown to be directly applicable to more general tasks such as age and head-pose estimation. Code is available at https://github.com/fregu856/ebms_regression.

## 1 Introduction

Supervised regression entails learning a model capable of predicting a continuous target value $y$ from an input $x$, given a set of paired training examples. It is a fundamental machine learning problem with many important applications within computer vision and other domains. Common regression tasks within computer vision include object detection [47, 23, 28, 63], head- and body-pose estimation [5, 57, 52, 59], age estimation [48, 42, 4], visual tracking [38, 64, 31, 8] and medical image registration [39, 6], just to mention a few. Today, such regression problems are commonly tackled using Deep Neural Networks (DNNs), due to their ability to learn powerful feature representations directly from data.

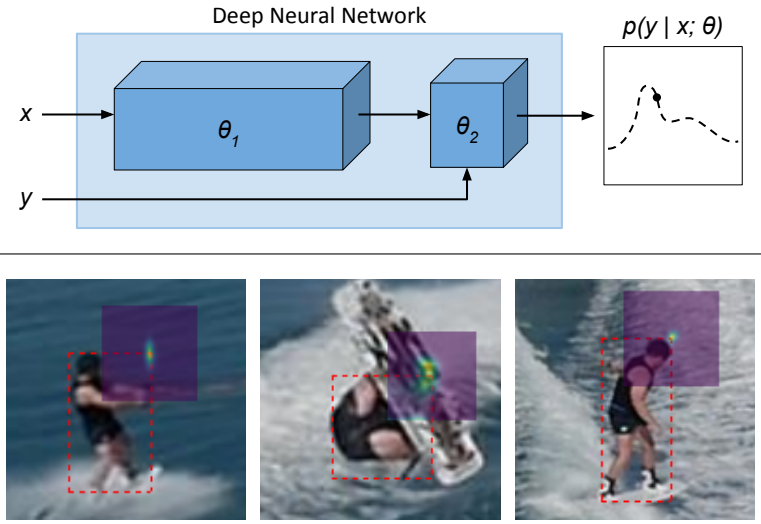



**Fig. 1.** An overview of the proposed regression method (top). We train an energy-based model $p(y|x;\theta) \propto e^{f_\theta(x,y)}$ of the conditional target density $p(y|x)$, using a DNN $f_\theta$ to predict the un-normalized density directly from the input-target pair $(x, y)$. Our approach is capable of predicting highly flexible densities and produce highly accurate estimates. This is demonstrated for the problem of bounding box regression (bottom), visualizing the marginal density for the top right box corner as a heatmap.

While classification is generally addressed using standardized losses and output representations, a wide variety of different techniques are employed for regression. The most conventional strategy is to train a DNN to directly predict a target $y$ given an input $x$ [27]. In such *direct regression* approaches, the model parameters of the DNN are learned by minimizing a loss function, for example the $L^2$ or $L^1$ loss, penalizing discrepancy between the predicted and ground truth target values. From a probabilistic perspective, this approach corresponds to creating a simple parametric model of the conditional target density $p(y|x)$, and minimizing the associated negative log-likelihood. The $L^2$ loss, for example, corresponds to a fixed-variance Gaussian model. More recent work [24, 26, 7, 15, 54, 45] has also explored learning more expressive models of $p(y|x)$, by letting a DNN instead output the full set of parameters of a certain family of probability distributions. To allow for straightforward implementation and training, many of these *probabilistic regression* approaches however restrict the parametric model to unimodal distributions such as Gaussian [26, 7] or Laplace [24, 15, 22], still severely limiting the expressiveness of the learned conditional target density. While these methods benefit from a clear probabilistic interpretation, they thus fail to fully exploit the predictive power of the DNN.

The quest for improved regression accuracy has also led to the development of more specialized methods, designed for a specific set of tasks. In computer vision, one particularly popular approach is that of *confidence-based regression*. Here,

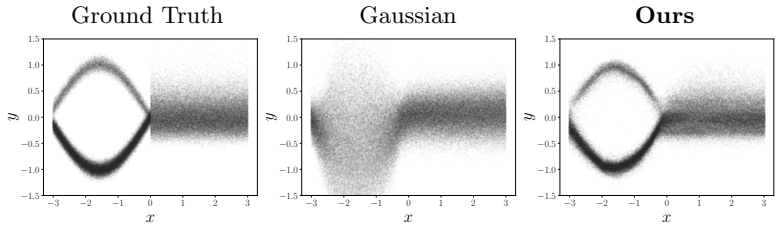



**Fig. 2.** An illustrative 1D regression problem. The training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{2000}$ is generated by the ground truth conditional target density $p(y|x)$. Our energy-based model $p(y|x;\theta) \propto e^{f_\theta(x,y)}$ of $p(y|x)$ is trained by directly minimizing the associated negative log-likelihood, approximated using Monte Carlo importance sampling. In contrast to the Gaussian model $p(y|x;\theta) = \mathcal{N}(y; \mu_\theta(x), \sigma_\theta^2(x))$, our energy-based model can learn multimodal and complex conditional target densities directly from data.

a DNN instead predicts a scalar confidence value for input-target pairs $(x, y)$. The confidence can then be maximized w.r.t. $y$ to obtain a target prediction for a given input $x$. This approach is commonly employed for image-coordinate regression tasks within e.g. human pose estimation [5, 57, 52] and object detection [28, 63], where a 2D heatmap over image pixel coordinates $y$ is predicted. Recently, the approach was also applied to the problem of bounding box regression by Jiang et al. [23]. Their proposed method, IoU-Net, obtained state-of-the-art accuracy on object detection, and was later also successfully applied to the task of visual tracking [8]. The training of such confidence-based regression methods does however entail generating additional pseudo ground truth labels, e.g. by employing a Gaussian kernel [55, 57], and selecting an appropriate loss function. This both requires numerous design choices to be made, and limits the general applicability of the methods. Moreover, confidence-based regression methods do not allow for a natural probabilistic interpretation in terms of the conditional target density $p(y|x)$. In this work, we therefore set out to develop a method combining the general applicability and the clear interpretation of probabilistic regression with the predictive power of the confidence-based approaches.

**Contributions** We propose a general and conceptually simple regression method with a clear probabilistic interpretation. Our method employs an energy-based model [30] to predict the un-normalized conditional target density $p(y|x)$ from the input-target pair $(x, y)$. It is trained by directly minimizing the associated negative log-likelihood, exploiting tailored Monte Carlo approximations. At test time, targets are predicted by maximizing the conditional target density $p(y|x)$ through gradient-based refinement. Our energy-based model is straightforward both to implement and train. Unlike commonly used probabilistic models, it can however still learn highly flexible target densities directly from data, as visualized in Figure 2. Compared to confidence-based approaches, our method requires no pseudo labels, benefits from a clear probabilistic interpretation, and is directly applicable to a variety of computer vision applications. We evaluate the proposed method on four diverse computer vision regression tasks: object

detection, visual tracking, age estimation and head-pose estimation. Our method is found to significantly outperform both direct regression baselines, and popular probabilistic and confidence-based alternatives, including the state-of-the-art IoU-Net [23]. Notably, our method achieves a 2.2% AP improvement over FPN Faster-RCNN [32] when applied for object detection on COCO [33], and sets a new state-of-the-art on standard benchmarks [37, 36] when applied for bounding box estimation in the recent ATOM [8] visual tracker. Our method is also shown to be directly applicable to the more general tasks of age and head-pose estimation, consistently improving performance of a variety of baselines.

## 2 Background & Related Work

In supervised regression, the task is to learn to predict a target value $y^\star \in \mathcal{Y}$ from a corresponding input $x^\star \in \mathcal{X}$, given a training set of i.i.d. input-target examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x, y)$. As opposed to classification, the target space $\mathcal{Y}$ is a continuous set, e.g. $\mathcal{Y} = \mathbb{R}^K$. In computer vision, the input space $\mathcal{X}$ often corresponds to the space of images, whereas the output space $\mathcal{Y}$ depends on the task at hand. Common examples include $\mathcal{Y} = \mathbb{R}^2$ in image-coordinate regression [57, 28], $\mathcal{Y} = \mathbb{R}_+$ in age estimation [48, 42], and $\mathcal{Y} = \mathbb{R}^4$ in object bounding box regression [47, 23]. A variety of techniques have previously been applied to supervised regression tasks. In order to motivate and provide intuition for our proposed method, we here describe a few popular approaches.

**Direct Regression** Over the last decade, DNNs have been shown to excel at a wide variety of regression problems. Here, a DNN is viewed as a function $f_\theta : \mathcal{U} \to \mathcal{O}$, parameterized by a set of learnable weights $\theta \in \mathbb{R}^P$. The most conventional regression approach is to train a DNN to directly predict the targets, $y^\star = f_\theta(x^\star)$, called *direct regression*. The model parameters $\theta$ are learned by minimizing a loss $\ell(f_\theta(x_i), y_i)$ that penalizes discrepancy between the prediction $f_\theta(x_i)$ and the ground truth target value $y_i$ on training examples $(x_i, y_i)$. Common choices include the $L^2$ loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_2^2$, the $L^1$ loss, $\ell(\hat{y}, y) = \|\hat{y} - y\|_1$, and their close relatives [21, 27]. From a probabilistic perspective, the choice of loss corresponds to minimizing the negative log-likelihood $-\log p(y|x;\theta)$ for a specific model $p(y|x;\theta)$ of the conditional target density. For example, the $L^2$ loss is derived from a fixed-variance Gaussian model, $p(y|x;\theta) = \mathcal{N}(y; f_\theta(x), \sigma^2)$.

**Probabilistic Regression** More recent work [24, 26, 7, 15, 22, 34, 54] has explicitly taken advantage of this probabilistic perspective to achieve more flexible parametric models $p(y|x;\theta) = p(y;\phi_\theta(x))$, by letting the DNN output the parameters $\phi$ of a family of probability distributions $p(y;\phi)$. For example, a general 1D Gaussian model can be realized as $p(y|x;\theta) = \mathcal{N}\big(y; \mu_\theta(x), \sigma_\theta^2(x)\big)$, where the DNN outputs the mean and log-variance as $f_\theta(x) = \phi_\theta(x) = [\,\mu_\theta(x)\ \log\sigma_\theta^2(x)\,]^\mathsf{T} \in \mathbb{R}^2$. The model parameters $\theta$ are learned by minimizing the negative log-likelihood $-\sum_{i=1}^N \log p(y_i|x_i;\theta)$ over the training set $\mathcal{D}$. At test time, a target estimate $y^\star$ is obtained by first predicting the density parameter values $\phi_\theta(x^\star)$ and then, for instance, taking the expected value of $p(y;\phi_\theta(x))$. Previous work has applied simple Gaussian and Laplace models on computer vision tasks such as object

detection [13, 19] and optical flow estimation [15, 22], usually aiming to not only achieve accurate predictions, but also to provide an estimate of aleatoric uncertainty [24, 17]. To allow for multimodal models $p(y; \phi_\theta(x))$, mixture density networks (MDNs) [3] have also been applied [34, 54]. The DNN then outputs weights for $K$ mixture components along with $K$ sets of parameters, e.g. $K$ sets of means and log-variances for a mixture of Gaussians. Previous work has also applied *infinite* mixture models by utilizing the conditional VAE (cVAE) framework [51, 45]. A latent variable model $p(y|x;\theta) = \int p(y; \phi_\theta(x,z))p(z; \phi_\theta(x))dz$ is then employed, where $p(y; \phi_\theta(x,z))$ and $p(z; \phi_\theta(x)$ typically are Gaussian distributions. Our proposed method also entails predicting a conditional target density $p(y|x;\theta)$ and minimizing the associated negative log-likelihood. However, our energy-based model $p(y|x;\theta)$ is not limited to the functional form of any specific probability density (e.g. Gaussian or Laplace), but is instead directly defined via a learned scalar function of $(x, y)$. In contrast to MDNs and cVAEs, our model $p(y|x;\theta)$ is not even limited to densities which are simple to generate samples from. This puts *minimal restricting assumptions* on the true $p(y|x)$, allowing it to be efficiently learned directly from data.

**Confidence-Based Regression** Another category of approaches reformulates the regression problem as $y^\star = \arg\max_y f_\theta(x,y)$, where $f_\theta(x,y) \in \mathbb{R}$ is a scalar confidence value predicted by the DNN. The idea is thus to predict a quantity $f_\theta(x,y)$, depending on both input $x$ and target $y$, that can be maximized over $y$ to obtain the final prediction $y^\star$. This maximization-based formulation is inherent in Structural SVMs [53], but has also been adopted for DNNs. We term this family of approaches *confidence-based regression*. Compared to direct regression, the predicted confidence $f_\theta(x,y)$ can encapsulate multiple hypotheses and other ambiguities. Confidence-based regression has been shown particularly suitable for image-coordinate regression tasks, such as hand keypoint localization [50] and body-part detection [55, 44, 57]. In these cases, a CNN is trained to output a 2D heatmap over the image pixel coordinates $y$, thus taking full advantage of the translational invariance of the problem. In computer vision, confidence prediction has also been successfully employed for tasks other than pure image-coordinate regression. Jiang et al. [23] proposed the IoU-Net for bounding box regression in object detection, where a bounding box $y \in \mathbb{R}^4$ and image $x$ are both input to the DNN to predict a confidence $f_\theta(x,y)$. It employs a pooling-based architecture that is differentiable w.r.t. the bounding box $y$, allowing efficient gradient-based maximization to obtain the final estimate $y^\star = \arg\max_y f_\theta(x,y)$. IoU-Net was later also successfully applied to target object estimation in visual tracking [8].

In general, confidence-based approaches are trained using a set of *pseudo label* confidences $c(x_i, y_i, y)$ generated for each training example $(x_i, y_i)$, and by employing a loss $\ell\big(f_\theta(x_i,y), c(x_i,y_i,y)\big)$. One strategy [44, 28] is to treat the confidence prediction as a binary classification problem, where $c(x_i,y_i,y)$ represents either the class, $c \in \{0,1\}$, or its probability, $c \in [0,1]$, and employ cross-entropy based losses $\ell$. The other approach is to treat the confidence prediction as a direct regression problem itself by applying standard regression losses, such as $L^2$ [50, 8, 55] or the Huber loss [23]. In these cases, the pseudo label confi-

dences $c$ can be constructed using a similarity measure $S$ in the target space, $c(x_i, y_i, y) = S(y, y_i)$, for example defined as the Intersection over Union (IoU) between two bounding boxes [23] or simply by a Gaussian kernel [55, 57, 52].

While these methods have demonstrated impressive results, confidence-based approaches thus require important design choices. In particular, the strategy for constructing the pseudo labels $c$ and the choice of loss $\ell$ are often crucial for performance and highly *task-dependent*, limiting general applicability. Moreover, the predicted confidence $f_\theta(x, y)$ can be difficult to interpret, since it has no natural connection to the conditional target density $p(y|x)$. In contrast, our approach is directly trained to predict $p(y|x)$ itself, and importantly it does *not* require generation of pseudo label confidences or choosing a specific loss.

**Regression-by-Classification** A regression problem can also be treated as a classification problem by first discretizing the target space $\mathcal{Y}$ into a finite set of $C$ classes. Standard techniques from classification, such as softmax and the cross-entropy loss, can then be employed. This approach has previously been applied to both age estimation [48, 42, 60] and head-pose estimation [49, 59]. The discretization of the target space $\mathcal{Y}$ however complicates exploiting its inherent neighborhood structure, an issue that has been addressed by exploring ordinal regression methods for 1D problems [4, 10]. While our energy-based approach can be seen as a generalization of the softmax model for classification to the continuous target space $\mathcal{Y}$, it does not suffer from the aforementioned drawbacks of regression-by-classification. On the contrary, our model naturally allows the network to exploit the full structure of the continuous target space $\mathcal{Y}$.

**Energy-Based Models** Our approach is of course also related to the theoretical framework of energy-based models, which often has been employed for machine learning problems in the past [35, 20, 30]. It involves learning an energy function $\mathcal{E}_\theta(x) \in \mathbb{R}$ that assigns low energy to observed data $x_i$ and high energy to other values of $x$. Recently, energy-based models have been used primarily for unsupervised learning problems within computer vision [58, 14, 11, 29, 40], where DNNs are directly used to predict $\mathcal{E}_\theta(x)$. These models are commonly trained by minimizing the negative log-likelihood, stemming from the probabilistic model $p(x; \theta) = e^{-\mathcal{E}_\theta(x)} / \int e^{-\mathcal{E}_\theta(x)} dx$, for example by generating approximate image samples from $p(x; \theta)$ using Markov Chain Monte Carlo [14, 11, 40]. In contrast, we study the application of energy-based models for $p(y|x)$ in *supervised* regression, a mostly overlooked research direction in recent years, and obtain state-of-the-art performance on four diverse computer vision regression tasks.

## 3 Proposed Regression Method

We propose a general and conceptually simple regression method with a clear probabilistic interpretation. Our method employs an energy-based model within a probabilistic regression formulation, defined in Section 3.1. In Section 3.2, we introduce our training strategy which is designed to be simple, yet highly effective and applicable to a wide variety of regression tasks within computer vision. Lastly, we describe our prediction strategy for high accuracy in Section 3.3.

### 3.1 Formulation

We take the probabilistic view of regression by creating a model $p(y|x;\theta)$ of the conditional target density $p(y|x)$, in which $\theta$ is learned by minimizing the associated negative log-likelihood. Instead of defining $p(y|x;\theta)$ by letting a DNN predict the parameters of a certain family of probability distributions (e.g. Gaussian or Laplace), we construct a versatile energy-based model that can better leverage the predictive power of DNNs. To that end, we take inspiration from confidence-based regression approaches and let a DNN directly predict a scalar value for any input-target pair $(x, y)$. Unlike confidence-based methods however, this prediction has a clear probabilistic interpretation. Specifically, we view a DNN as a function $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, parameterized by $\theta \in \mathbb{R}^P$, that maps an input-target pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to a scalar value $f_\theta(x, y) \in \mathbb{R}$. Our model $p(y|x;\theta)$ of the conditional target density $p(y|x)$ is then defined according to,

$$p(y|x;\theta) = \frac{e^{f_\theta(x,y)}}{Z(x,\theta)}, \qquad Z(x,\theta) = \int e^{f_\theta(x,\tilde{y})} d\tilde{y}\,, \tag{1}$$

where $Z(x,\theta)$ is the input-dependent normalizing partition function. We train this energy-based model (1) by directly minimizing the negative log-likelihood $-\log p(\{y_i\}_i|\{x_i\}_i;\theta) = \sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$, where each term is given by,

$$-\log p(y_i|x_i;\theta) = \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i, y_i). \tag{2}$$

This direct and straightforward training approach thus requires the evaluation of the generally intractable $Z(x,\theta) = \int e^{f_\theta(x,y)} dy$. Many fundamental computer vision tasks, such as object detection, keypoint estimation and pose estimation, however rely on regression problems with a low-dimensional target space $\mathcal{Y}$. In such cases, effective finite approximations of $Z(x,\theta)$ can be applied. In some tasks, such as image-coordinate regression, this is naturally performed by a grid approximation, utilizing the dense prediction obtained by fully-convolutional networks. In this work, we however investigate a more *generally applicable* technique, namely Monte Carlo approximations with importance sampling. This procedure, when employed for training the network, is detailed in Section 3.2.

At test time, given an input $x^\star$, our model in (1) allows evaluating the conditional target density $p(y|x^\star;\theta)$ for any target $y$ by first approximating $Z(x^\star,\theta)$, and then predicting the scalar $f_\theta(x^\star, y)$ using the DNN. This enables the computation of, e.g., the mean and variance of the target value $y$. In this work, we take inspiration from confidence-based regression and focus on finding the most likely prediction, $y^\star = \arg\max_y p(y|x^\star;\theta) = \arg\max_y f_\theta(x^\star, y)$, which does not require the evaluation of $Z(x^\star,\theta)$ during inference. Thanks to the auto-differentiation capabilities of modern deep learning frameworks, we can apply gradient-based techniques to find the final prediction by simply maximizing the network output $f_\theta(x^\star, y)$ w.r.t. $y$. We elaborate on this procedure for prediction in Section 3.3.

### 3.2 Training

Our model $p(y|x;\theta) = e^{f_\theta(x,y)}/Z(x,\theta)$ of the conditional target density is trained by directly minimizing the negative log-likelihood $\sum_{i=1}^{N} -\log p(y_i|x_i;\theta)$. To evaluate the integral in (2), we employ Monte Carlo importance sampling. Each term $-\log p(y_i|x_i;\theta)$ is therefore approximated by sampling values $\{y^{(k)}\}_{k=1}^{M}$ from a proposal distribution $q(y|y_i)$ that depends on the ground truth target value $y_i$,

$$-\log p(y_i|x_i;\theta) \approx \log\left(\frac{1}{M}\sum_{k=1}^{M}\frac{e^{f_\theta(x_i,y^{(k)})}}{q(y^{(k)}|y_i)}\right) - f_\theta(x_i,y_i). \tag{3}$$

The final loss $J(\theta)$ used to train the DNN $f_\theta$ is then obtained by averaging over all training examples $\{(x_i,y_i)\}_{i=1}^{n}$ in the current mini-batch,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i,y^{(i,m)})}}{q(y^{(i,m)}|y_i)}\right) - f_\theta(x_i,y_i), \tag{4}$$

where $\{y^{(i,m)}\}_{m=1}^{M}$ are $M$ samples drawn from $q(y|y_i)$. Qualitatively, minimizing $J(\theta)$ encourages the DNN to output large values $f_\theta(x_i,y_i)$ for the ground truth target $y_i$, while minimizing the predicted value $f_\theta(x_i,y)$ at all other targets $y$. In ambiguous or uncertain cases, the DNN can output small values everywhere or large values at multiple hypotheses, but at the cost of a higher loss.

As can be seen in (4), the DNN $f_\theta$ is applied both to the input-target pair $(x_i,y_i)$, and all input-sample pairs $\{(x_i,y^{(i,m)})\}_{m=1}^{M}$ during training. While this can seem inefficient, most applications in computer vision employ network architectures that first extract a deep feature representation for the input $x_i$. The DNN $f_\theta$ can thus be designed to combine this input feature with the target $y$ at a late stage, as visualized in Figure 1. The input feature extraction process, which becomes the main computational bottleneck, therefore needs to be performed only once for each $x_i$. In practice, we found our training strategy to not add any significant overhead compared to the direct regression baselines, and the computational cost to be *identical* to that of the confidence-based methods.

Compared to confidence-based regression, a significant advantage of our approach is however that there is no need for generating task-dependent pseudo label confidences or choosing between different losses. The *only* design choice of our training method is the proposal distribution $q(y|y_i)$. Note however that since the loss $J(\theta)$ in (4) explicitly adapts to $q(y|y_i)$, this choice has no effect on the overall behaviour of the loss, only on the quality of the sampled approximation. We found a mixture of a few equally weighted Gaussian components, all centered at the target label $y_i$, to consistently perform well in our experiments across all four diverse computer vision applications. Specifically, $q(y|y_i)$ is set to,

$$q(y|y_i) = \frac{1}{L}\sum_{l=1}^{L}\mathcal{N}(y;y_i,\sigma_l^2 I), \tag{5}$$

where the standard deviations $\{\sigma_l\}_{l=1}^{L}$ are hyperparameters selected based on a validation set for each experiment. We only considered the simple Gaussian

proposal in (5), as this was found sufficient to obtain state-of-the-art experimental results. Full ablation studies for the number of components $L$ and $\{\sigma_l\}_{l=1}^{L}$ are provided in the supplementary material. Figure 2 illustrates that our model $p(y|x;\theta)$ can learn complex conditional target densities, containing both multi-modalities and asymmetry, directly from data using the described training procedure. In this illustrative example, we use (5) with $L = 2$ and $\sigma_1 = 0.1$, $\sigma_2 = 0.8$.

### 3.3 Prediction

Given an input $x^\star$ at test time, the trained DNN $f_\theta$ can be used to evaluate the full conditional target density $p(y|x^\star;\theta) = e^{f_\theta(x^\star,y)}/Z(x^\star,\theta)$, by employing the aforementioned techniques to approximate the constant $Z(x^\star,\theta)$. In many applications, the most likely prediction $y^\star = \arg\max_y p(y|x^\star;\theta)$ is however the single desired output. For our energy-based model, this is obtained by directly maximizing the DNN output, $y^\star = \arg\max_y f_\theta(x^\star,y)$, thus not requiring $Z(x^\star,\theta)$ to be evaluated. By taking inspiration from IoU-Net [23] and designing the DNN $f_\theta$ to be differentiable w.r.t. the target $y$, the gradient $\nabla_y f_\theta(x^\star,y)$ can be efficiently evaluated using the auto-differentiation tools implemented in modern deep learning frameworks. An estimate of $y^\star = \arg\max_y f_\theta(x^\star,y)$ can therefore be obtained by performing gradient ascent to find a local maximum of $f_\theta(x^\star,y)$.

The gradient ascent refinement is performed either on a single initial estimate $\hat{y}$, or on a set of random initializations $\{\hat{y}_k\}_{k=1}^{K}$ to obtain a final accurate prediction $y^\star$. Starting at $y = \hat{y}_k$, we thus run $T$ gradient ascent iterations, $y \leftarrow y + \lambda\nabla_y f_\theta(x^\star,y)$, with step-length $\lambda$. In our experiments, we fix $T$ (typically, $T = 10$) and select $\lambda$ using grid search on a validation set. As noted in Section 3.2, this prediction procedure can be made highly efficient by extracting the feature representation for $x^\star$ only once. Back-propagation is then performed only through a few final layers of the DNN to evaluate the gradient $\nabla_y f_\theta(x^\star,y)$. The gradient computation for a set of candidates $\{\hat{y}_k\}_{k=1}^{K}$ can also be parallelized on the GPU by simple batching, requiring no significant overhead. Overall, the inference speed is somewhat decreased compared to direct regression baselines, but is *identical* to confidence-based methods such as IoU-Net [23]. An algorithm detailing this prediction procedure is found in the supplementary material.

## 4 Experiments

We perform comprehensive experiments on four different computer vision regression tasks: object detection, visual tracking, age estimation and head-pose estimation. Our proposed approach is compared both to baseline regression methods and to state-of-the-art models. Notably, our method significantly outperforms the confidence-based IoU-Net [23] method for bounding box regression in direct comparisons, both when applied for object detection on the COCO dataset [33] and for target object estimation in the recent ATOM [8] visual tracker. On age and head-pose estimation, our approach is shown to consistently improve performance of a variety of baselines. All experiments are implemented in PyTorch [43]. For all tasks, further details are also provided in the supplementary material.

**Table 1.** Impact of $L$ and $\{\sigma_l\}_{l=1}^{L}$ in the proposal distribution $q(y|y_i)$ (5), for the object detection task on the *2017 val* split of the COCO [33] dataset. For $L = 2$, $\sigma_1 = \sigma_2/4$. For $L = 3$, $\sigma_1 = \sigma_3/4$ and $\sigma_2 = \sigma_3/2$. $L = 3$ with $\sigma_L = 0.15$ is selected.

| Number of components $L$ | 1 | | | 2 | | | 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Base proposal st. dev. $\sigma_L$ | 0.02 | 0.04 | 0.08 | 0.1 | 0.15 | 0.2 | 0.1 | 0.15 | 0.2 |
| AP (%) | 38.1 | 38.5 | 37.5 | 39.0 | **39.1** | 39.0 | 39.0 | **39.1** | 38.8 |

**Table 2.** Results for the object detection task on the *2017 test-dev* split of the COCO [33] dataset. Our proposed method significantly outperforms the baseline FPN Faster-RCNN [32] and the state-of-the-art confidence-based IoU-Net [23].

| Formulation | Direct | Gaussian | Gaussian | Gaussian | Gaussian | Gaussian | Laplace | Confidence | Confidence | |
|---|---|---|---|---|---|---|---|---|---|---|
| Approach | Faster-RCNN | | Mixt. 2 | Mixt. 4 | Mixt. 8 | cVAE | | IoU-Net | 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 |

### 4.1 Object Detection

We first perform experiments on object detection, the task of classifying and estimating a bounding box for each object in a given image. Specifically, we compare our regression method to other techniques for the task of bounding box regression, by integrating them into an existing object detection pipeline. To this end, we use the Faster-RCNN [47] framework, which serves as a popular baseline in the object detection field due to its strong state-of-the-art performance. It employs one network head for classification and one head for regressing the bounding box using the direct regression approach. We also include various probabilistic regression baselines and compare with simple Gaussian and Laplace models, by modifying the Faster-RCNN regression head to predict both the mean and log-variance of the distribution, and adopting the associated negative log-likelihood loss. Similarly, we compare with mixtures of $K = \{2, 4, 8\}$ Gaussians by duplicating the modified regression head $K$ times and adding a network head for predicting $K$ component weights. Moreover, we compare with an infinite mixture of Gaussians by training a cVAE. Finally, we also compare our approach to the state-of-the-art confidence-based IoU-Net [23]. It extends Faster-RCNN with an additional branch that predicts the IoU overlap between a target bounding box $y$ and the ground truth. The IoU prediction branch uses differentiable region pooling [23], allowing the initial bounding box predicted by the Faster-RCNN to be refined using gradient-based maximization of the predicted IoU confidence.

For our approach, we employ an *identical architecture* as used in IoU-Net for a fair comparison. Instead of training the network to output the IoU, we predict the exponent $f_\theta(x, y)$ in (1), trained by minimizing the negative log-likelihood in (4). We parametrize the bounding box as $y = (c_x/w_0, c_y/h_0, \log w, \log h) \in \mathbb{R}^4$, where $(c_x, c_y)$ and $(w, h)$ denote the center coordinate and size, respectively. The reference size $(w_0, h_0)$ is set to that of the ground truth during training and the initial box during prediction. Based on the ablation study found in Table 1, we

**Table 3.** Results for the visual tracking task on the two common datasets TrackingNet [37] and UAV123 [36]. The symbol † indicates an approximate value (±1), taken from the plot in the corresponding paper. Our proposed method significantly outperforms the baseline ATOM and other recent state-of-the-art trackers.

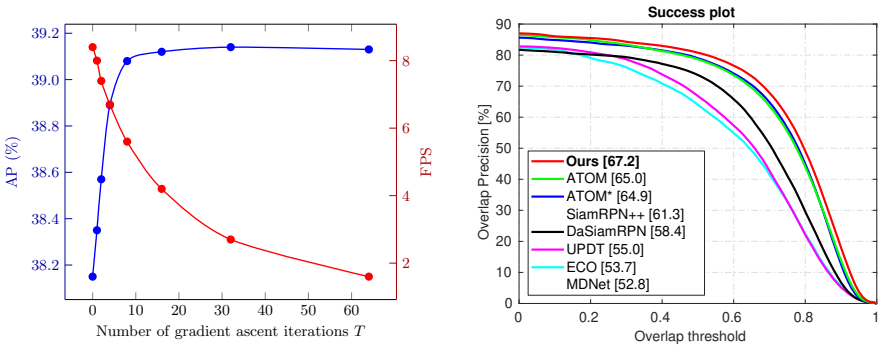
| Dataset | Metric | ECO [9] | SiamFC [1] | MDNet [38] | UPDT [2] | DaSiamRPN [64] | SiamRPN++ [31] | ATOM [8] | ATOM* | **Ours** |
|---|---|---|---|---|---|---|---|---|---|---|
| TrackingNet | Precision (%) | 49.2 | 53.3 | 56.5 | 55.7 | 59.1 | 69.4 | 64.8 | 66.6 | **69.7** |
| | Norm. Prec. (%) | 61.8 | 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 | $OP_{0.50}$ (%) | 64.0 | - | - | 66.8 | 73.6 | 75† | 78.9 | 79.0 | **80.8** |
| | $OP_{0.75}$ (%) | 32.8 | - | - | 32.9 | 41.1 | 56† | 55.7 | 56.5 | **60.2** |
| | AUC (%) | 53.7 | - | 52.8 | 55.0 | 58.4 | 61.3 | 65.0 | 64.9 | **67.2** |

employ $L = 3$ isotropic Gaussians with standard deviation $\sigma_l = 0.0375 \cdot 2^{l-1}$ for the proposal distribution (5). In addition to the standard IoU-Net, we compare with a version (denoted IoU-Net*) employing the same proposal distribution and inference settings as in our approach. For both our method and IoU-Net*, we set the refinement step-length $\lambda$ using grid search on a separate validation set.

Our experiments are performed on the large-scale COCO benchmark [33]. We use the *2017 train* split (≈ 118 000 images) for training and the *2017 val* split (≈ 5 000 images) for setting our hyperparameters. The results are reported on the *2017 test-dev* split (≈ 20 000 images), in terms of the standard COCO metrics AP, $AP_{50}$ and $AP_{75}$. We also report the inference speed in terms of frames-per-second (FPS) on a single NVIDIA TITAN Xp GPU. We initialize all networks in our comparison with the pre-trained Faster-RCNN weights, using the ResNet50-FPN [32] backbone, and re-train *only* the newly added layers for a fair comparison. The results are shown in Table 2. Our proposed method obtains the best results, significantly outperforming Faster-RCNN and IoU-Net by 2.2% and 1.1% in AP, respectively. The Gaussian model is outperformed by the mixture of 2 Gaussians, but note that adding more components does *not* further improve the performance. In comparison, the cVAE achieves somewhat improved performance, but is still clearly outperformed by our method. Compared to the probabilistic baselines, we believe that our energy-based model offers a more direct and effective representation of the underlying density via the scalar DNN output $f_\theta(x, y)$. The inference speed of our method is somewhat lower than that of Faster-RCNN, but identical to IoU-Net. How the number of iterations $T$ in the gradient-based refinement affects inference speed and performance is analyzed in Figure 3a, showing that our choice $T = 10$ provides a reasonable trade-off.

### 4.2 Visual Tracking

Next, we evaluate our approach on the problem of generic visual object tracking. The task is to estimate the bounding box of a target object in every frame of a video. The target object is defined by a given box in the first video frame. We employ the recently introduced ATOM [8] tracker as our baseline. Given the first-frame annotation, ATOM trains a classifier to first roughly localize the target in a new frame. The target bounding box is then determined using




(a) (b)

**Fig. 3. (a)** Impact of the number of gradient ascent iterations $T$ on performance (AP) and inference speed (FPS), for the object detection task on the *2017 val* split of the COCO [33] dataset. **(b)** Success plot on the UAV123 [36] visual tracking dataset, showing the overlap precision $\text{OP}_H$ as a function of the overlap threshold $H$.

an IoU-Net-based module, which is also conditioned on the first-frame target appearance using a modulation-based architecture. We train our network to predict the conditional target density through $f_\theta(x, y)$ in (1), using a network architecture *identical* to the baseline ATOM tracker. In particular, we employ the same bounding box parameterization as for object detection (Section 4.1) and sample $M = 128$ boxes during training from a proposal distribution (5) generated by $L = 2$ Gaussians with standard deviations $\sigma_1 = 0.05$, $\sigma_2 = 0.5$. During tracking, we follow the same procedure as in ATOM, sampling 10 boxes in each frame followed by gradient ascent to refine the estimate generated by the classification module. The inference speed of our approach is thus identical to ATOM, running at over 30 FPS on a single NVIDIA GT-1080 GPU.

We demonstrate results on two standard tracking benchmarks: TrackingNet [37] and UAV123 [36]. TrackingNet contains challenging videos sampled from YouTube, with a test set of 511 videos. The main metric is the Success, defined as the average IoU overlap with the ground truth. UAV123 contains 123 videos captured from a UAV, and includes small and fast-moving objects. We report the overlap precision metric ($\text{OP}_H$), defined as the percentage of frames having bounding box IoU overlap larger than a threshold $H$. The final AUC score is computed as the average OP over all thresholds $H \in [0, 1]$. Hyperparameters are set on the OTB [56] and NFS [25] datasets, containing 100 videos each. Due to the significant challenges imposed by the limited supervision and generic nature of the tracking problem, there are no competitive baselines employing direct bounding box regression. Current state-of-the-art employ either confidence-based regression, as in ATOM, or anchor-based bounding box regression techniques [64, 31]. We therefore only compare with the ATOM baseline and include other recent state-of-the-art methods in the comparison. As in Section 4.1, we also

**Table 4.** Results for the age estimation task on the UTKFace [62] dataset. Gradient-based refinement using our proposed method consistently improves MAE (lower is better) for the age predictions produced by a variety of different baselines.

| +Refine | Niu et al. [41] | Cao et al. [4] | Direct | Gaussian | Laplace | Softmax (CE, $L^2$) | Softmax (CE, $L^2$, Var) |
|---|---|---|---|---|---|---|---|
| | 5.74 ± 0.05 | 5.47 ± 0.01 | 4.81 ± 0.02 | 4.79 ± 0.06 | 4.85 ± 0.04 | 4.78 ± 0.05 | 4.81 ± 0.03 |
| ✓ | - | - | **4.65** ± 0.02 | 4.66 ± 0.04 | 4.81 ± 0.04 | **4.65** ± 0.04 | 4.69 ± 0.03 |

compare with a version (denoted ATOM*) of the IoU-Net-based ATOM employing the same training and inference settings as our final approach. The results are shown in Table 3, and the success plot on UAV123 is shown in Figure 3b. Our approach achieves a significant 2.5% and 2.2% absolute improvement over ATOM on the overall metric on TrackingNet and UAV123, respectively. Note that the improvements are most prominent for high-accuracy boxes, as indicated by $\text{OP}_{0.75}$. Our approach also outperforms the recent SiamRPN++ [31], which employs anchor-based bounding box regression [47, 46] and a much deeper backbone network (ResNet50) compared to ours (ResNet18). Figure 1 (bottom) visualizes an illustrative example of the target density $p(y|x;\theta) \propto e^{f_\theta(x,y)}$ predicted by our approach during tracking. As illustrated, it predicts flexible densities which qualitatively capture meaningful uncertainty in challenging cases.

### 4.3 Age Estimation

To demonstrate the general applicability of our proposed method, we also perform experiments on regression tasks not involving bounding boxes. In age estimation, we are given a cropped image $x \in \mathbb{R}^{h\times w\times 3}$ of a person's face, and the task is to predict his/her age $y \in \mathbb{R}_+$. We utilize the UTKFace [62] dataset, specifically the subset of 16 434 images used by Cao et al. [4]. We also utilize the dataset split employed in [4], with 3 287 test images and 11 503 images for training. Additionally, we use 1 644 of the training images for validation. Methods are evaluated in terms of the Mean Absolute Error (MAE). The DNN architecture $f_\theta(x, y)$ of our model first extracts ResNet50 [18] features $g_x \in \mathbb{R}^{2048}$ from the input image $x$. The age $y$ is processed by four fully-connected layers, generating $g_y \in \mathbb{R}^{128}$. The two feature vectors are then concatenated and processed by two fully-connected layers, outputting $f_\theta(x, y) \in \mathbb{R}$. We apply our proposed method to refine the age predicted by baseline models, using the gradient ascent maximization of $f_\theta(x, y)$ detailed in Section 3.3. All baseline DNN models employ a similar architecture, including an identical ResNet50 for feature extraction and the same number of fully-connected layers to output either the age $y \in \mathbb{R}$ (*Direct*), mean and variance parameters for Gaussian and Laplace distributions, or to output logits for $C$ discretized classes (*Softmax*). The results are found in Table 4. We observe that age refinement provided by our method consistently improves the accuracy of the predictions generated by the baselines. For *Direct*, e.g., this refinement marginally decreases inference speed from 49 to 36 FPS.

**Table 5.** Results for the head-pose estimation task on the BIWI [12] dataset. Gradient-based refinement using our proposed method consistently improves the average MAE (lower is better) for yaw, pitch and roll for the predicted pose produced by our baselines.

| +Refine | Gu et al. [16] | Yang et al. [59] | Direct | Gaussian | Laplace | Softmax (CE, $L^2$) | Softmax (CE, $L^2$, Var) |
|---|---|---|---|---|---|---|---|
| | 3.66 | 3.60 | 3.09 ± 0.07 | 3.12 ± 0.08 | 3.21 ± 0.06 | 3.04 ± 0.08 | 3.15 ± 0.07 |
| ✓ | - | - | 3.07 ± 0.07 | 3.11 ± 0.07 | 3.19 ± 0.06 | **3.01** ± 0.07 | 3.11 ± 0.06 |

### 4.4 Head-Pose Estimation

Lastly, we evaluate our method on the task of head-pose estimation. In this case, we are given an image $x \in \mathbb{R}^{h \times w \times 3}$ of a person, and the aim is to predict the orientation $y \in \mathbb{R}^3$ of his/her head, where $y$ is the yaw, pitch and roll angles. We utilize the BIWI [12] dataset, specifically the processed dataset provided by Yang et al. [59], in which the images have been cropped to faces detected using MTCNN [61]. We also employ protocol 2 as defined in [59], with 10 613 images for training and 5 065 images for testing. Additionally, we use 1 010 training images for validation. The methods are evaluated in terms of the average MAE for yaw, pitch and roll. The network architecture of the DNN $f_\theta(x, y)$ defining our model takes the image $x \in \mathbb{R}^{h \times w \times 3}$ and orientation $y \in \mathbb{R}^3$ as inputs, but is otherwise identical to the age estimation case (Section 4.3). Our model is again evaluated by applying the gradient-based refinement to the predicted orientation $y \in \mathbb{R}^3$ produced by a number of baseline models. We use the same baselines as for age estimation, and apart from minor changes required to increase the output dimension from 1 to 3, identical network architectures are also used. The results are found in Table 5, and also in this case we observe that refinement using our proposed method consistently improves upon the baselines.

## 5 Conclusion

We proposed a general and conceptually simple regression method with a clear probabilistic interpretation. It models the conditional target density $p(y|x)$ by predicting the un-normalized density through a DNN $f_\theta(x, y)$, taking the input-target pair $(x, y)$ as input. This energy-based model $p(y|x; \theta) = e^{f_\theta(x,y)}/Z(x, \theta)$ of $p(y|x)$ is trained by directly minimizing the associated negative log-likelihood, employing Monte Carlo importance sampling to approximate the partition function $Z(x, \theta)$. At test time, targets are predicted by maximizing the DNN output $f_\theta(x, y)$ w.r.t. $y$ via gradient-based refinement. Extensive experiments performed on four diverse computer vision tasks demonstrate the high accuracy and wide applicability of our method. Future directions include exploring improved architectural designs, studying other regression applications, and investigating our proposed method's potential for aleatoric uncertainty estimation.

**Acknowledgments** This research was supported by the Swedish Foundation for Strategic Research via *ASSEMBLE*, the Swedish Research Council via *Learning flexible models for nonlinear dynamics*, the ETH Zürich Fund (OK), a Huawei Technologies Oy (Finland) project, an Amazon AWS grant, and Nvidia.

## References


1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision (ECCV) Workshops (2016) 11
2. Bhat, G., Johnander, J., Danelljan, M., Shahbaz Khan, F., Felsberg, M.: Unveiling the power of deep tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 483–498 (2018) 11
3. Bishop, C.M.: Mixture density networks (1994) 5
4. Cao, W., Mirjalili, V., Raschka, S.: Rank-consistent ordinal regression for neural networks. arXiv preprint arXiv:1901.07884 (2019) 1, 6, 13
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7291–7299 (2017) 1, 3
6. Chou, C.R., Frederick, B., Mageras, G., Chang, S., Pizer, S.: 2D/3D image registration using regression learning. Computer Vision and Image Understanding **117**(9), 1095–1106 (2013) 1
7. Chua, K., Calandra, R., McAllister, R., Levine, S.: Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 4759–4770 (2018) 2, 4
8. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: Accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4660–4669 (2019) 1, 3, 4, 5, 9, 11
9. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: Efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6638–6646 (2017) 11
10. Diaz, R., Marathe, A.: Soft labels for ordinal regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 6
11. Du, Y., Mordatch, I.: Implicit generation and modeling with energy based models. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 6
12. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L.: Random forests for real time 3d face analysis. International Journal of Computer Vision (IJCV) **101**(3), 437–458 (2013) 14
13. Feng, D., Rosenbaum, L., Timm, F., Dietmayer, K.: Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3D object detection. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 1280–1287. IEEE (2019) 5
14. Gao, R., Lu, Y., Zhou, J., Zhu, S.C., Nian Wu, Y.: Learning generative convnets via multi-grid modeling and sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9155–9164 (2018) 6
15. Gast, J., Roth, S.: Lightweight probabilistic deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3369–3378 (2018) 2, 4, 5
16. Gu, J., Yang, X., De Mello, S., Kautz, J.: Dynamic facial analysis: From bayesian filtering to recurrent neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1548–1557 (2017) 14
17. Gustafsson, F.K., Danelljan, M., Schön, T.B.: Evaluating scalable Bayesian deep learning methods for robust computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020) 5

18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 13
19. He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2888–2897 (2019) 5
20. Hinton, G., Osindero, S., Welling, M., Teh, Y.W.: Unsupervised discovery of nonlinear structure using contrastive backpropagation. Cognitive science **30**(4), 725–731 (2006) 6
21. Huber, P.J.: Robust estimation of a location parameter. The Annals of Mathematical Statistics pp. 73–101 (1964) 4
22. Ilg, E., Cicek, O., Galesso, S., Klein, A., Makansi, O., Hutter, F., Bro, T.: Uncertainty estimates and multi-hypotheses networks for optical flow. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 652–667 (2018) 2, 4, 5
23. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 784–799 (2018) 1, 3, 4, 5, 6, 9, 10
24. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems (NeurIPS). pp. 5574–5584 (2017) 2, 4, 5
25. Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: A benchmark for higher frame rate object tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1125–1134 (2017) 12
26. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6402–6413 (2017) 2, 4
27. Lathuilière, S., Mesejo, P., Alameda-Pineda, X., Horaud, R.: A comprehensive analysis of deep regression. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2019) 2, 4
28. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018) 1, 3, 4, 5
29. Lawson, D., Tucker, G., Dai, B., Ranganath, R.: Energy-inspired models: Learning with sampler-induced distributions. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) 6
30. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting structured data **1**(0) (2006) 3, 6
31. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4282–4291 (2019) 1, 11, 12, 13
32. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125 (2017) 4, 10, 11
33. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 740–755 (2014) 4, 9, 10, 11, 12

34. Makansi, O., Ilg, E., Cicek, O., Brox, T.: Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7144–7153 (2019) 4, 5
35. Mnih, A., Hinton, G.: Learning nonlinear constraints with contrastive backpropagation. In: Proceedings of the IEEE International Joint Conference on Neural Networks. vol. 2, pp. 1302–1307. IEEE (2005) 6
36. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 445–461 (2016) 4, 11, 12
37. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 300–317 (2018) 4, 11, 12
38. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4293–4302 (2016) 1, 11
39. Niethammer, M., Huang, Y., Vialard, F.X.: Geodesic regression for image time-series. In: International conference on medical image computing and computer-assisted intervention. pp. 655–662. Springer (2011) 1
40. Nijkamp, E., Hill, M., Han, T., Zhu, S.C., Wu, Y.N.: On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In: Thirty-Fourth AAAI Conference on Artificial Intelligence (2020) 6
41. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Ordinal regression with multiple output cnn for age estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4920–4928 (2016) 13
42. Pan, H., Han, H., Shan, S., Chen, X.: Mean-variance loss for deep age estimation from a face. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5285–5294 (2018) 1, 4, 6
43. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 8024–8035 (2019) 9
44. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: DeepCut: Joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4929–4937 (2016) 5
45. Prokudin, S., Gehler, P., Nowozin, S.: Deep directional statistics: Pose estimation with uncertainty quantification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 534–551 (2018) 2, 5
46. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7263–7271 (2017) 13
47. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **39**, 1137–1149 (2015) 1, 4, 10, 13
48. Rothe, R., Timofte, R., Van Gool, L.: Deep expectation of real and apparent age from a single image without facial landmarks. International Journal of Computer Vision (IJCV) **126**(2-4), 144–157 (2016) 1, 4, 6

49. Ruiz, N., Chong, E., Rehg, J.M.: Fine-grained head pose estimation without keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2074–2083 (2018) 6
50. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR). pp. 1145–1153 (2017) 5
51. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 3483–3491 (2015) 5
52. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5693–5703 (2019) 1, 3, 6
53. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research (JMLR) **6**(Sep), 1453–1484 (2005) 5
54. Varamesh, A., Tuytelaars, T.: Mixture dense regression for object detection and human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13086–13095 (2020) 2, 4, 5
55. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4732 (2016) 3, 5, 6
56. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **37**(9), 1834–1848 (2015) 12
57. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 466–481 (2018) 1, 3, 4, 5, 6
58. Xie, J., Lu, Y., Zhu, S.C., Wu, Y.: A theory of generative convnet. In: International Conference on Machine Learning (ICML). pp. 2635–2644 (2016) 6
59. Yang, T.Y., Chen, Y.T., Lin, Y.Y., Chuang, Y.Y.: 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). pp. 1087–1096 (2019) 1, 6, 14
60. Yang, T.Y., Huang, Y.H., Lin, Y.Y., Hsiu, P.C., Chuang, Y.Y.: SSR-Net: A compact soft stagewise regression network for age estimation. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2018) 6
61. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016) 14
62. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5810–5818 (2017), https://susanqq.github.io/UTKFace/ 13
63. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 850–859 (2019) 1, 3
64. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 101–117 (2018) 1, 11, 12