How to Train Your Energy-Based Model for Regression

Fredrik K. Gustafsson[1]
fredrik.gustafsson@it.uu.se
Martin Danelljan[2]
martin.danelljan@vision.ee.ethz.ch
Radu Timofte[2]
radu.timofte@vision.ee.ethz.ch
Thomas B. Schön[1]
thomas.schon@it.uu.se

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

Abstract

Energy-based models (EBMs) have become increasingly popular within computer vision in recent years. While they are commonly employed for generative image modeling, recent work has applied EBMs also for regression tasks, achieving state-of-the-art performance on object detection and visual tracking. Training EBMs is however known to be challenging. While a variety of different techniques have been explored for generative modeling, the application of EBMs to regression is not a well-studied problem. How EBMs should be trained for best possible regression performance is thus currently unclear. We therefore accept the task of providing the first detailed study of this problem. To that end, we propose a simple yet highly effective extension of noise contrastive estimation, and carefully compare its performance to six popular methods from literature on the tasks of 1D regression and object detection. The results of this comparison suggest that our training method should be considered the go-to approach. We also apply our method to the visual tracking task, achieving state-of-the-art performance on five datasets. Notably, our tracker achieves 63.7% AUC on LaSOT and 78.7% Success on TrackingNet. Code is available at https://github.com/fregu856/ebms_regression.

1 Introduction

Energy-based models (EBMs) [27] have a rich history in machine learning [2, 16, 34, 41, 49]. An EBM specifies a probability density $p(x;\theta) = e^{f_\theta(x)} / \int e^{f_\theta(x)} dx$ directly via a parameterized scalar function $f_\theta(x)$. By defining $f_\theta(x)$ using a deep neural network (DNN), $p(x;\theta)$ becomes expressive enough to learn practically any density from observed data. EBMs have therefore become increasingly popular within computer vision in recent years, commonly being applied for various generative image modeling tasks [9, 11, 12, 13, 39, 40, 55].

Recent work [8, 14] has also explored conditional EBMs as a general formulation for regression, demonstrating particularly impressive performance on the tasks of object detection [26, 44, 57] and visual tracking [9, 7, 28]. Regression entails predicting a continuous target y from an input x, given a training set of observed input-target pairs. This was addressed

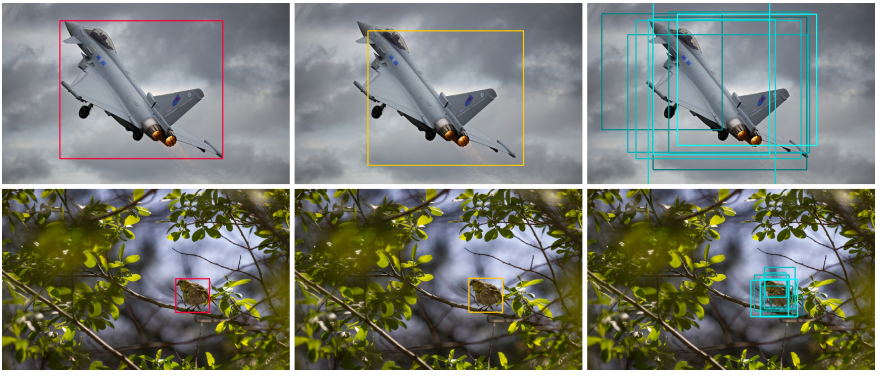

Figure 1: We propose NCE+ to train EBMs $p(y|x;\theta)$ for tasks such as bounding box regression. NCE+ is a highly effective extension of NCE, accounting for noise in the annotation process of real-world datasets. Given a label y_i (red box), the EBM is trained by having to discriminate between $y_i + \nu_i$ (yellow box) and noise samples $\{y^{(i,m)}\}_{m=1}^{M}$ (blue boxes).

in [8, 14] by learning a conditional EBM $p(y|x;\theta)$, capturing the distribution of the target value y given the input x. At test time, gradient ascent was then used to maximize $p(y|x;\theta)$ w.r.t. y, producing highly accurate predictions. Regression is a fundamental problem within computer vision with many additional applications [25, 42, 45, 54, 56], which all would benefit from such accurate predictions. In this work, we therefore study the use of EBMs for regression in detail, aiming to further improve its performance and applicability.

While the modeling capacity of EBMs makes them highly attractive for many applications, training EBMs is known to be challenging. This is because the EBM $p(x;\theta) = e^{f_\theta(x)} / \int e^{f_\theta(x)} dx$ involves an intractable integral, complicating the use of standard maximum likelihood (ML) learning. A variety of different techniques have therefore been explored in the generative modeling literature, including alternative estimation methods [12, 15, 20, 48, 50] and approximations based on Markov chain Monte Carlo (MCMC) [9, 17, 39, 40]. The application of EBMs for regression is however not a particularly well-studied problem. [8, 14] both applied importance sampling to approximate intractable integrals, an approach known to scale poorly with the data dimensionality, and considered no alternative techniques. How EBMs $p(y|x;\theta)$ should be trained for best possible performance on computer vision regression tasks is thus an open question, which we set out to investigate in this work.

Contributions We propose a simple yet highly effective extension of noise contrastive estimation (NCE) [15] to train EBMs $p(y|x;\theta)$ for regression tasks. Our proposed method, termed *NCE+*, can be understood as a direct generalization of NCE, accounting for noise in the annotation process. We evaluate NCE+ on illustrative 1D regression problems and on the task of bounding box regression in object detection. We also provide a detailed comparison of NCE+ and *six* popular methods from previous work, the results of which suggest that NCE+ should be considered the go-to training method. Lastly, we apply our proposed NCE+ to the task of visual tracking, achieving state-of-the-art results on *five* common datasets.

2 Energy-Based Models for Regression

We study the application of EBMs to important regression tasks in computer vision, using energy-based models of the conditional density $p(y|x)$. Here, we first define the general

regression problem and our employed EBM in Section 2.1. Our prediction strategy based on gradient ascent is then described in Section 2.2. Lastly, we discuss the challenges associated with training EBMs, and describe six popular methods from the literature, in Section 2.3.

2.1 Problem & Model Definition

In a supervised regression problem, we are given a training set $\mathcal{D}$ of i.i.d. input-target pairs, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $(x_i, y_i) \sim p(x,y)$. The task is then to learn how to predict a target $y^\star \in \mathcal{Y}$ given a new input $x^\star \in \mathcal{X}$. The target space $\mathcal{Y}$ is continuous, $\mathcal{Y} = \mathbb{R}^K$ for some $K \geq 1$, and the input space $\mathcal{X}$ usually corresponds to the space of images.

As in [8, 14], we address this problem by creating an energy-based model $p(y|x;\theta)$ of the conditional target density $p(y|x)$. To that end, we specify a DNN $f_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ with model parameters $\theta \in \mathbb{R}^P$. This DNN directly maps any input-target pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$ to a scalar $f_\theta(x,y) \in \mathbb{R}$. The model $p(y|x;\theta)$ of the conditional target density is then defined as,

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

where the DNN output $f_\theta(x,y) \in \mathbb{R}$ is interpreted as the negative energy of the density, and $Z(x,\theta)$ is the input-dependent normalizing partition function. Since $p(y|x;\theta)$ in (1) is directly defined by the DNN f_θ, minimal restricting assumptions are put on the true $p(y|x)$. The predictive power of the DNN can thus be fully exploited, enabling learning of, *e.g.*, multimodal and asymmetric densities directly from data. This expressivity however comes at the cost of $Z(x,\theta)$ being intractable, which complicates evaluating or sampling from $p(y|x;\theta)$.

2.2 Prediction

At test time, the problem of predicting a target value $y^\star$ from an input $x^\star$ corresponds to finding a point estimate of the predicted conditional density $p(y|x^\star;\theta)$. The most natural choice is to select the most likely target under the model, $y^\star = \arg\max_y p(y|x^\star;\theta) = \arg\max_y f_\theta(x^\star,y)$. The prediction $y^\star$ is thus obtained by directly maximizing the DNN scalar output $f_\theta(x^\star,y)$ w.r.t. y, not requiring $Z(x^\star,\theta)$ to be evaluated nor any samples from $p(y|x^\star;\theta)$ to be generated. Following [8, 14], we estimate $y^\star = \arg\max_y f_\theta(x^\star,y)$ by performing gradient ascent to refine an initial estimate $\hat{y}$ and find a local maximum of $f_\theta(x^\star,y)$. Starting at $y = \hat{y}$, we thus run T gradient ascent iterations, $y \leftarrow y + \lambda \nabla_y f_\theta(x^\star,y)$, with step-length λ. An algorithm for this prediction procedure is found in the supplementary material.

2.3 Training

To train the DNN $f_\theta(x,y)$ specifying the EBM (1), different techniques for fitting a density $p(y|x;\theta)$ to observed data $\{(x_i,y_i)\}_{i=1}^N$ can be used. In general, the most commonly applied such technique is ML learning, which entails minimizing the negative log-likelihood (NLL),

$$-\sum_{i=1}^N \log p(y_i|x_i;\theta) = \sum_{i=1}^N \log\left(\int e^{f_\theta(x_i,y)} dy\right) - f_\theta(x_i,y_i), \tag{2}$$

w.r.t. the parameters θ. The integral in (2) is however intractable, and exact evaluation of the NLL is thus not possible. [8, 14] employed importance sampling to approximate such intractable integrals, obtaining state-of-the-art performance on object detection and visual

tracking. Recent work [9, 12, 13, 29, 40, 46] on generative image modeling has however applied a variety of different training methods not considered in [8, 14], including the ML learning alternatives NCE [15] and score matching [20]. How we should train the DNN f_θ to obtain best possible regression performance is thus unclear. In this work, we therefore carefully compare our proposed method to six popular training methods from the literature.

ML with Importance Sampling (ML-IS) A straightforward training method is proposed in [14], which we term *ML with Importance Sampling (ML-IS)*. Using ML-IS, [14] successfully applied the EBM (1) to the regression tasks of object detection, visual tracking, age estimation and head-pose estimation. In ML-IS, the DNN f_θ is trained by directly minimizing the NLL (2) w.r.t. θ, using importance sampling to approximate the intractable integral,

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

Here, $\{y^{(i,m)}\}_{m=1}^{M}$ are M samples drawn from a proposal distribution $q(y|y_i)$ that depends on the ground truth target y_i. In [14], $q(y|y_i)$ is set to a mixture of K Gaussians centered at y_i,

$$q(y|y_i) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y;y_i,\sigma_k^2 I). \tag{4}$$

The loss $J(\theta)$ is obtained by averaging over all pairs $\{(x_i,y_i)\}_{i=1}^{n}$ in the current mini-batch,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i,y^{(i,m)})}}{q(y^{(i,m)}|y_i)}\right) - f_\theta(x_i,y_i). \tag{5}$$

KL Divergence with Importance Sampling (KLD-IS) Instead of minimizing the NLL (2), [8] considers the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(p(y|y_i) \parallel p(y|x_i;\theta))$ between the EBM $p(y|x_i;\theta)$ and an assumed density $p(y|y_i)$ of the true target y given the label y_i. The density $p(y|y_i)$ models noise in the annotation process of our given training set $\mathcal{D} = \{(x_i,y_i)\}_{i=1}^{N}$. In [8], $p(y|y_i) = \mathcal{N}(y;y_i,\sigma^2 I)$, where σ is a hyperparameter. As shown in [8],

$$D_{\mathrm{KL}}(p(y|y_i) \parallel p(y|x_i;\theta)) = \log\left(\int e^{f_\theta(x_i,y)}dy\right) - \int f_\theta(x_i,y)p(y|y_i)dy + C, \tag{6}$$

where C is a constant that does not depend on θ. [8] approximates the integrals in (6) using importance sampling, employing the ML-IS proposal $q(y|y_i)$ in (4). By then averaging over all pairs $\{(x_i,y_i)\}_{i=1}^{n}$ in the current mini-batch, the loss $J(\theta)$ used to train f_θ is obtained as,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{1}{M}\sum_{m=1}^{M}\frac{e^{f_\theta(x_i,y^{(i,m)})}}{q(y^{(i,m)}|y_i)}\right) - \frac{1}{M}\sum_{m=1}^{M} f_\theta(x_i,y^{(i,m)})\frac{p(y^{(i,m)}|y_i)}{q(y^{(i,m)}|y_i)}, \tag{7}$$

where $\{y^{(i,m)}\}_{m=1}^{M}$ are M samples drawn from the proposal $q(y|y_i)$. We term this training method *KL Divergence with Importance Sampling (KLD-IS)*. When applied to visual tracking in [8], KLD-IS outperformed ML-IS and set a new state-of-the-art.

ML with MCMC (ML-MCMC) To minimize the NLL (2) w.r.t. the parameters θ, the following identity for the expression of the gradient $\nabla_\theta - \log p(y_i|x_i;\theta)$ can be utilized [27],

$$\nabla_\theta - \log p(y_i|x_i;\theta) = \mathbb{E}_{p(y|x_i;\theta)}\Big[\nabla_\theta f_\theta(x_i,y)\Big] - \nabla_\theta f_\theta(x_i,y_i). \tag{8}$$

The expectation in (8) is then approximated using samples $\{y^{(i,m)}\}_{m=1}^{M}$ drawn from $p(y|x_i;\theta)$, *i.e.* from the EBM itself. To obtain each sample $y^{(i,m)} \sim p(y|x_i;\theta)$, MCMC is used. Specifically, we follow recent work [9, 11, 13, 39, 40, 55] on generative image modeling and run $L \geq 1$ steps of Langevin dynamics [52]. Starting at $y_{(0)}$, we thus update $y_{(l)}$ according to,

$$y_{(l+1)} = y_{(l)} + \frac{\alpha^2}{2}\nabla_y f_\theta(x_i, y_{(l)}) + \alpha\varepsilon_l, \quad \varepsilon_l \sim \mathcal{N}(0, I), \tag{9}$$

and set $y^{(i,m)} = y_{(L)}$. Here, $\alpha > 0$ is a small constant step-length. Following the principle of contrastive divergence [17, 27, 49], we start the Markov chain (9) at the ground truth target, $y_{(0)} = y_i$. By approximating (8) with the samples $\{y^{(i,m)}\}_{m=1}^{M}$, and by averaging over all pairs $\{(x_i, y_i)\}_{i=1}^{n}$ in the current mini-batch, the loss $J(\theta)$ used to train the DNN f_θ is obtained as,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{M}\sum_{m=1}^{M} f_\theta(x_i, y^{(i,m)})\right) - f_\theta(x_i, y_i). \tag{10}$$

We term this specific training method *ML with MCMC (ML-MCMC)*.

Noise Contrastive Estimation (NCE) As an alternative to ML learning, Gutmann and Hyvärinen proposed NCE [15] for estimating unnormalized parametric models. NCE entails generating samples from some noise distribution p_N, and learning to discriminate between these noise samples and observed data examples. It has recently been applied to generative image modeling with EBMs [12], and the NCE loss is also utilized in various frameworks for self-supervised learning [1, 5, 18]. Moreover, NCE has been applied to train EBMs for supervised *classification* tasks within language modeling [22, 32, 33, 35], where the target space $\mathcal{Y}$ is a large but finite set of possible labels. We adopt NCE for regression by using a noise distribution $p_N(y|y_i)$ of the same form as the ML-IS proposal in (4),

$$p_N(y|y_i) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I), \tag{11}$$

and by employing the ranking NCE objective [22], as described in [32]. We choose ranking NCE over the binary objective since it is consistent under a weaker assumption [32]. We thus define $y^{(i,0)} \triangleq y_i$, and train the DNN f_θ by minimizing the following loss,

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\exp\left\{f_\theta(x_i, y^{(i,0)}) - \log p_N(y^{(i,0)}|y_i)\right\}}{\sum_{m=0}^{M}\exp\left\{f_\theta(x_i, y^{(i,m)}) - \log p_N(y^{(i,m)}|y_i)\right\}}, \tag{12}$$

where $\{y^{(i,m)}\}_{m=1}^{M}$ are M noise samples drawn from $p_N(y|y_i)$ in (11).

Score Matching (SM) Another alternative estimation method is score matching (SM), as proposed by Hyvärinen [20] and further studied for supervised problems in [47]. The method focuses on the *score* of $p(y|x;\theta)$, defined as $\nabla_y \log p(y|x;\theta) = \nabla_y f_\theta(x,y)$, aiming for it to approximate the score of the true target density $p(y|x)$. Note that the EBM score $\nabla_y f_\theta(x,y)$ does not depend on the intractable $Z(x,\theta)$. SM was applied to simple conditional density estimation problems in [47], using a combination of feed-forward networks and reproducing kernels to specify the EBM. Following [47], we train the DNN f_θ by minimizing the loss,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{tr}\left(\nabla_y^2 f_\theta(x_i, y_i)\right) + \frac{1}{2}\left\|\nabla_y f_\theta(x_i, y_i)\right\|_2^2, \tag{13}$$

where only the diagonal of $\nabla_y^2 f_\theta(x_i, y_i)$ actually is needed to compute the first term.

Denoising Score Matching (DSM) By modifying the SM objective, denoising score matching (DSM) was proposed by Vincent [50]. DSM does not require computation of any second derivatives, improving its scalability to high-dimensional data. The method entails employing SM on noise-corrupted data points. Recently, DSM has been successfully applied to generative image modeling [29, 46, 48]. DSM was also extended to train EBMs of conditional densities in [23], where it was applied to a transfer learning problem. Following [23], we use a Gaussian noise distribution and train the DNN f_θ by minimizing the loss,

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{M}\sum_{m=1}^{M}\left\| \nabla_y f_\theta(x_i, \tilde{y}^{(i,m)}) + \frac{\tilde{y}^{(i,m)} - y_i}{\sigma^2} \right\|_2^2, \tag{14}$$

where $\{\tilde{y}^{(i,m)}\}_{m=1}^M$ are M samples drawn from the noise distribution $p_\sigma(\tilde{y}|y_i) = \mathcal{N}(\tilde{y}; y_i, \sigma^2 I)$.

3 Proposed Training Method

To train the DNN f_θ specifying our EBM $p(y|x;\theta)$ in (1), we propose a *simple yet highly effective* extension of NCE [15]. Motivated by the improved performance of KLD-IS compared to ML-IS on visual tracking [8], we extend NCE with the capability to model annotation noise. To that end, we adopt the standard NCE noise distribution p_N (11) and loss (12), but instead of defining $y^{(i,0)} \triangleq y_i$, we sample $v_i \sim p_\beta(y)$ and define $y^{(i,0)} \triangleq y_i + v_i$. The distribution p_β is a zero-centered version of p_N in which $\{\sigma_k\}_{k=1}^K$ are scaled with $\beta > 0$,

$$p_N(y|y_i) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; y_i, \sigma_k^2 I), \qquad p_\beta(y) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y; 0, \beta\sigma_k^2 I). \tag{15}$$

Instead of training the DNN f_θ by learning to discriminate between noise samples $\{y^{(i,m)}\}_{m=1}^M$ and the label y_i, it thus has to discriminate between the samples $\{y^{(i,m)}\}_{m=1}^M$ and $y_i + v_i$. Examples of $y_i + v_i$ and $\{y^{(i,m)}\}_{m=1}^M$ in the task of bounding box regression are visualized in Figure 1. Similar to KLD-IS, in which an assumed density of the true target value y given y_i is employed, our approach thus accounts for possible noise and inaccuracies in the provided label y_i. Specifically, our proposed training method entails sampling $\{y^{(i,m)}\}_{m=1}^M \sim p_N(y|y_i)$ and $v_i \sim p_\beta(y)$, setting $y^{(i,0)} \triangleq y_i + v_i$, and minimizing the following loss,

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\exp\left\{f_\theta(x_i, y^{(i,0)}) - \log p_N(y^{(i,0)}|y_i)\right\}}{\sum_{m=0}^{M}\exp\left\{f_\theta(x_i, y^{(i,m)}) - \log p_N(y^{(i,m)}|y_i)\right\}}. \tag{16}$$

As $\beta \to 0$, samples $v_i \sim p_\beta(y)$ will concentrate increasingly close to zero, and the standard NCE method is in practice recovered. Our proposed training method can thus be understood as a direct generalization of NCE. Compared to NCE, our method adds no significant training cost and requires tuning of a single additional hyperparameter β. A value for β is selected in a simple two-step procedure. First, we fix $y^{(i,0)} = y_i$ and select the standard deviations $\{\sigma_k\}_{k=1}^K$ based on validation set performance, just as in NCE. We then fix $\{\sigma_k\}_{k=1}^K$ and vary β to find the value corresponding to maximum validation performance. Typically, we start this ablation with $\beta = 0.1$. We term our proposed training method *NCE+*.

	ML-IS	ML-MCMC-1	ML-MCMC-16	ML-MCMC-256	KLD-IS	NCE	SM	DSM	**NCE+**
$D_{\mathrm{KL}} \downarrow$	**0.062**	0.865	0.449	0.106	0.088	0.068	0.781	0.395	0.066
Training Cost $\downarrow$	**0.44**	0.54	2.41	30.8	**0.44**	0.45	0.60	0.47	0.46

Table 1: Comparison of training methods for the illustrative 1D regression experiments.

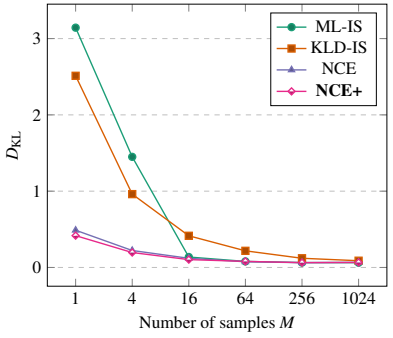

Figure 2: Detailed comparison of the top-performing methods for the illustrative 1D regression experiments. NCE and NCE+ here demonstrate clear superior performance for small number of samples M.

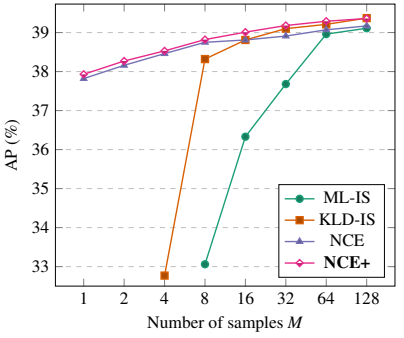

Figure 3: Detailed comparison of the top-performing methods for object detection, on the *2017 val* split of COCO [30]. Missing values for ML-IS and KLD-IS correspond to failed training due to numerical issues.

4 Comparison of Training Methods

We provide a detailed comparison of the six training methods from Section 2.3 and our proposed NCE+. To that end, we perform extensive experiments on 1D regression (Section 4.1) and object detection (Section 4.2). Our findings are summarized in Section 4.3. All experiments are implemented in PyTorch [43] and the code is publically available. For both tasks, further details and results are also provided in the supplementary material.

4.1 1D Regression Experiments

We first perform experiments on illustrative 1D regression problems. The DNN $f_\theta(x,y)$ is here a simple feed-forward network, taking $x \in \mathbb{R}$ and $y \in \mathbb{R}$ as inputs. We employ two synthetic datasets, and evaluate the training methods by how well the learned model $p(y|x;\theta)$ (1) approximates the known ground truth $p(y|x)$, as measured by the KL divergence D_{KL}.

Results A comparison of all seven training methods in terms of D_{KL} and training cost (seconds per epoch) is found in Table 1. For ML-MCMC, we include results for $L \in \{1,16,256\}$ Langevin steps (9). We observe that ML-IS, KLD-IS, NCE and NCE+ clearly have the best performance. While ML-MCMC is relatively close in terms of D_{KL} for $L = 256$, this comes at the expense of a massive increase in training cost. DSM outperforms SM in terms of both metrics, but is not close to the top-performing methods. The four best methods are further compared in Figure 2, showing D_{KL} as a function of M. Here, we observe that NCE and NCE+ significantly outperform ML-IS and KLD-IS for small number of samples M.

4.2 Object Detection Experiments

Next, we evaluate the methods on the task of bounding box regression in object detection. We employ an identical network architecture for $f_\theta(x,y)$ as in [14]. An extra network branch,

	ML-IS	ML-MCMC-1	ML-MCMC-4	ML-MCMC-8	KLD-IS	NCE	DSM	NCE+
AP (%) ↑	39.4	36.4	36.4	36.4	39.6	39.5	36.3	**39.7**
AP_{50}(%) ↑	58.6	57.9	57.9	58.0	58.6	58.6	57.9	**58.7**
AP_{75}(%) ↑	42.1	38.8	39.0	39.0	42.6	42.4	38.9	**42.7**
Training Cost ↓	1.03	2.47	7.05	13.3	**1.02**	1.04	3.84	1.09

Table 2: Comparison of training methods for the object detection experiments, on the *2017 test-dev* split of COCO [30]. Our proposed NCE+ achieves the best performance.

consisting of three fully-connected layers with parameters θ, is thus added onto a pre-trained and fixed FPN Faster-RCNN detector [31]. Given an image x and bounding box $y \in \mathbb{R}^4$, the image is first processed by the detector backbone network (ResNet50-FPN), outputting image features $h_1(x)$. Using a differentiable PrRoiPool [21] layer, $h_1(x)$ is then pooled to extract features $h_2(x,y)$. Finally, $h_2(x,y)$ is processed by the added network branch, outputting $f_\theta(x,y) \in \mathbb{R}$. As in [14], predictions $y^\star$ are produced by performing guided NMS [21] followed by gradient-based refinement (Section 2.2), taking the Faster-RCNN detections as initial estimates $\hat{y}$. Experiments are performed on the large-scale COCO dataset [30]. We use the *2017 train* split ($\approx$ 118 000 images) for training, the *2017 val* split ($\approx$ 5 000 images) for setting hyperparameters, and report results on the *2017 test-dev* split ($\approx$ 20 000 images). The standard COCO metrics AP, AP_{50} and AP_{75} are used, where AP is the primary metric.

Results A comparison of the training methods in terms of the COCO metrics and training cost (seconds per iteration) is found in Table 2. Since DSM clearly outperformed SM in the 1D regression experiments, we here only include DSM. For ML-MCMC, results for $L \in \{1,4,8\}$ are included. We observe that ML-IS, KLD-IS, NCE and NCE+ clearly have the best performance. In terms of the COCO metrics, NCE+ outperforms NCE and all other methods. ML-IS is also outperformed by KLD-IS. The four top-performing methods are further compared in Figure 3, in terms of AP as a function of the number of samples M. NCE and NCE+ here demonstrate clear superior performance for small values of M, and do not experience numerical issues even for $M=1$. KLD-IS improves this robustness compared ML-IS, but is not close to matching NCE or NCE+. In terms of training cost, the four top-performing methods are virtually identical. For ML-IS, *e.g.*, we observe in Figure 4 that setting $M=1$ decreases the training cost with 23% compared to the standard case of $M=128$.

Analysis of NCE+ Hyperparameters How the value of $\beta > 0$ in p_β (15) affects validation performance is studied in Figure 5. Here, we observe that quite a large range of values improve the performance compared to the NCE baseline ($\beta \to 0$), before it eventually degrades for $\beta \gtrsim 0.3$. We also observe that the performance is optimized for $\beta = 0.1$. In Figure 5, the standard deviations $\{\sigma_k\}_{k=1}^K$ in p_N, p_β (15) are set to $\{0.075, 0.15, 0.3\}$. These values are selected in an initial step based on an ablation study for NCE, which is found in Table 3.

4.3 Discussion

The results on both set of experiments are highly consistent. First of all, ML-IS, KLD-IS, NCE and NCE+ are by far the top-performing training methods. ML-MCMC, the method commonly employed for generative image modeling in recent years, does not come close to matching these top-performing methods, especially not given similar computational budgets. When studying the performance as a function of the number of samples M, NCE and NCE+ are the superior methods by a significant margin. In particular, this study demonstrates that the NCE and NCE+ losses are numerically more stable than those of ML-IS and KLD-IS. In

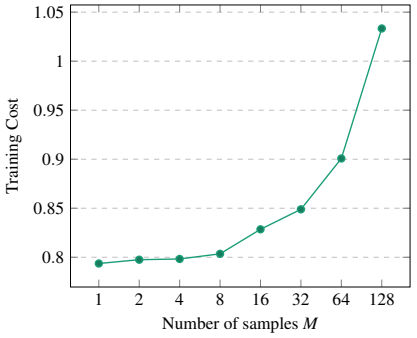

Figure 4: Effect of the number of samples M on training cost (seconds per iteration), for ML-IS on object detection.

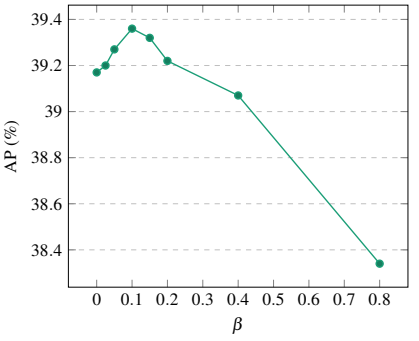

Figure 5: Effect of the NCE+ hyperparameter β on object detection performance (↑), on the *2017 val* split of COCO [30].

$\{\sigma_k\}_{k=1}^3$	{0.0125, 0.025, 0.05}	{0.025, 0.05, 0.1}	{0.05, 0.1, 0.2}	{0.075, 0.15, 0.3}	{0.1, 0.2, 0.4}
AP (%) ↑	38.58	38.95	39.12	**39.17**	39.05

Table 3: Ablation study for NCE, on the *2017 val* split of COCO [30].

the 1D regression problems, which employ synthetic datasets without any annotation noise, NCE and NCE+ have virtually identical performance. In the object detection experiments however, where we employ real-world datasets, NCE+ consistently improves the NCE performance. On object detection, NCE+ also improves or matches the performance of KLD-IS, which explicitly models annotation noise and outperforms ML-IS. Overall, the results of the comparison suggest that our proposed NCE+ should be considered the go-to training method.

5 Visual Tracking Experiments

Lastly, we apply our proposed NCE+ to the task of visual tracking. Specifically, we consider *generic visual object tracking*, which entails estimating the bounding box $y \in \mathbb{R}^4$ of a target object in every frame of a video. The target object does not belong to any pre-specified class, but is instead defined by a given bounding box in the initial video frame. We compare the performance both to NCE and KLD-IS, and to state-of-the-art trackers. Code and trained models are available at [6]. Further details are also found in the supplementary material.

Tracking Approach We base our tracker on the recent DiMP [4] and PrDiMP [8]. The target object is thus first coarsely localized in the current video frame via 2D image-coordinate regression of its center point, emphasizing robustness over accuracy. Then, the full bounding box $y \in \mathbb{R}^4$ of the target is accurately regressed by gradient-based refinement (Section 2.2). The two stages employ separate network branches which are trained jointly end-to-end. As a strong baseline, we combine the DiMP method for center point regression with the PrDiMP bounding box regression approach. We term this resulting tracker *DiMP-KLD-IS*. By also modifying common training parameters (batch size, data augmentation etc.), DiMP-KLD-IS significantly outperforms both DiMP and PrDiMP. Our proposed tracker, termed *DiMP-NCE+*, is then obtained simply by using NCE+ instead of KLD-IS to train the bounding box regression branch. In both cases, the number of samples $M = 128$. As in [4, 8], the training splits of TrackingNet [37], LaSOT [10], GOT-10k [19] and COCO [30] are used for training. Similar to PrDiMP, our DiMP-NCE+ tracker runs at about 30 FPS on a single GPU.

	MDNet [33]	UPDT [8]	DaSiamRPN [62]	ATOM [7]	SiamRPN++ [23]	DiMP [2]	SiamRCNN [51]	PrDiMP [8]	DiMP-KLD-IS	DiMP-NCE	**DiMP-NCE+**
TrackingNet	60.6	61.1	63.8	70.3	73.3	74.0	**81.2**	75.8	78.1	77.1	78.7
LaSOT	39.7	-	-	51.5	49.6	56.9	**64.8** (62.3)	59.8	63.1	62.8	63.7
UAV123	52.8	54.5	57.7	63.2	61.3	64.3	64.9	66.7	66.6	65.2	**67.2**
NFS	42.2	53.7	-	58.4	-	62.0	63.9	63.5	64.7	64.3	**65.0**
OTB-100	67.8	70.2	65.8	66.9	69.6	68.4	70.1 (68.0)	69.6	70.1	69.3	**70.7**

Table 4: Results for the visual tracking experiments. The AUC (Success) metric is reported on five common datasets. Our proposed DiMP-NCE+ tracker significantly outperforms strong baselines and achieves state-of-the-art performance on all five datasets. For SiamRCNN [51], results for the ResNet50 version are given in parentheses when available.

Results We evaluate DiMP-NCE+ on five commonly used tracking datasets. TrackingNet [37] is a large-scale dataset containing videos sampled from YouTube. Results are reported on its test set of 511 videos. We also evaluate on the LaSOT [10] test set, containing 280 long videos (2 500 frames on average). Moreover, we report results on the UAV123 [36] dataset, consisting of 123 videos which feature small targets and distractor objects. Results are also reported on the 30 FPS version of the need for speed (NFS) [24] dataset, containing 100 videos with fast motions. Finally, we evaluate on the 100 videos of OTB-100 [53]. Our tracker is evaluated in terms of overlap precision (OP). For a threshold $T \in [0,1]$, OP_T is the percentage of frames in which the IoU overlap between the estimated and ground truth target bounding box is larger than T. By averaging OP_T over $T \in [0,1]$, the *AUC* score is then obtained. For TrackingNet, the term *Success* is used in place of AUC. Results in terms of AUC on all five datasets are found in Table 4. To ensure significance, the average AUC over 5 runs is reported for our trackers. We observe that DiMP-NCE+ consistently outperforms both our DiMP-KLD-IS baseline, and a variant employing NCE instead of NCE+. Compared to previous approaches, only the very recent SiamRCNN [51] achieves results competitive with our DiMP-NCE+. SiamRCNN is however slower than DiMP-NCE+ (5 FPS vs 30 FPS) and employs a larger backbone network (ResNet101 vs ResNet50). Results for the ResNet50 version of SiamRCNN are only available on two of the datasets, on which it is outperformed by our DiMP-NCE+. More detailed results are provided in the supplementary material.

6 Conclusion

We proposed a simple yet highly effective extension of NCE to train EBMs $p(y|x;\theta)$ for computer vision regression tasks. Our proposed method NCE+ can be understood as a direct generalization of NCE, accounting for noise in the annotation process of real-world datasets. We also provided a detailed comparison of NCE+ and six popular methods from literature, the results of which suggest that NCE+ should be considered the go-to training method. This comparison is the first comprehensive study of how EBMs should be trained for best possible regression performance. Finally, we applied our proposed NCE+ to the task of visual tracking, achieving state-of-the-art performance on five commonly used datasets. We hope that our simple training method and promising results will encourage the research community to further explore the application of EBMs to various regression tasks.

Acknowledgments This research was financially supported by the Swedish Foundation for Strategic Research via the project *ASSEMBLE*, the Swedish Research Council via the project *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] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15509–15519, 2019.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[3] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–498, 2018.

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6182–6191, 2019.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[6] Martin Danelljan and Goutam Bhat. PyTracking: Visual tracking library based on PyTorch. https://github.com/visionml/pytracking, 2019. Accessed: 20/04/2020.

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4660–4669, 2019.

[8] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7183–7192, 2020.

[9] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5374–5383, 2019.

[11] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9155–9164, 2018.

[12] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. *arXiv preprint arXiv:1912.00589*, 2019.

[13] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2020.

[14] Fredrik K Gustafsson, Martin Danelljan, Goutam Bhat, and Thomas B Schön. Energy-based models for deep probabilistic regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.

[15] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 297–304, 2010.

[16] Geoffrey Hinton, Simon Osindero, Max Welling, and Yee-Whye Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006.

[17] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[20] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

[21] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[22] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[23] Ilyes Khemakhem, Ricardo Pio Monti, Diederik P Kingma, and Aapo Hyvärinen. ICE-BeeM: Identifiable conditional energy-based deep models. *arXiv preprint arXiv:2002.11537*, 2020.

[24] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1125–1134, 2017.

[25] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[26] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[27] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[28] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie 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.

[29] Zengyi Li, Yubei Chen, and Friedrich T Sommer. Learning energy-based models in high-dimensional spaces with multi-scale denoising score matching. *arXiv preprint arXiv:1910.07762*, 2019.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[32] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3698–3707, 2018.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119, 2013.

[34] Andriy Mnih and Geoffrey Hinton. Learning nonlinear constraints with contrastive backpropagation. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 2, pages 1302–1307. IEEE, 2005.

[35] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *International Conference on Machine Learning (ICML)*, pages 1751–1758, 2012.

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

[37] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard 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.

[38] Hyeonseob Nam and Bohyung 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.

[39] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5233–5243, 2019.

[40] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[41] Margarita Osadchy, Matthew L Miller, and Yann L Cun. Synergistic face detection and pose estimation with energy-based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1017–1024, 2005.

[42] Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5285–5294, 2018.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

[44] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[45] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2016.

[46] Saeed Saremi, Arash Mehrjou, Bernhard Schölkopf, and Aapo Hyvärinen. Deep energy estimator networks. *arXiv preprint arXiv:1805.08306*, 2018.

[47] Hiroaki Sasaki and Aapo Hyvärinen. Neural-kernelized conditional density estimation. *arXiv preprint arXiv:1806.01754*, 2018.

[48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11895–11907, 2019.

[49] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

[50] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[51] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6578–6588, 2020.

[52] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688, 2011.

[53] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1834–1848, 2015.

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

[55] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning (ICML)*, pages 2635–2644, 2016.

[56] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu 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.

[57] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 850–859, 2019.

[58] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.