# Learning with energy-based models

Thomas Pock

pock@icg.tugraz.at

Institute of Computer Graphics and Vision
Graz University of Technology

OWOS, 21/06/2021

# Energy-based models

- There is a renewed interest in energy based models[1]
- Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$, be a pair of output-input variables
- $y$ is a noisy image and $x$ is a denoised image
- $y$ is a set of stereo images, and $x$ is the disparity image
- In energy-based models, the task of inferring $x$ from $y$ is defined via an energy-minimization / optimization approach

$$\boxed{\hat{x} \in \operatorname*{argmin}_x E(x, y)}$$

- The energy $E(x, y)$ assigns a certain energy value to the configuration $(x, y)$
- Different algorithms for finding a minimizer (or at least a stationary point)
- Discrete / continuous / convex / non-convex / smooth / non-smooth ...

[1] CVPR 2021 Tutorial: "Theory and Application of Energy-Based Generative Models"

# Main properties

Energy-based models come along with many interesting properties:

- Energies can be hand-crafted based on first principles (physics-inspired)
- Energies can be learned from data using supervised, self-supervised or unsupervised learning
- Energies allow for multiple solutions $\hat{x} \in X$
- Characterization of the geometry of solutions via optimality conditions
- Energies $E(x, y)$ provide a quality measure for a particular candidate $x$
- Direct link to statistical modeling and Bayesian inference via $p(x|y) \propto \exp(-E(x, y))$
- Synthesis of samples from $p(x|y)$ via Langevin dynamics on E(x,y).

# Main properties

Energy-based models come along with many interesting properties:

- Energies can be hand-crafted based on first principles (physics-inspired)
- Energies can be learned from data using supervised, self-supervised or unsupervised learning
- Energies allow for multiple solutions $\hat{x} \in X$
- Characterization of the geometry of solutions via optimality conditions
- Energies $E(x, y)$ provide a quality measure for a particular candidate $x$
- Direct link to statistical modeling and Bayesian inference via $p(x|y) \propto \exp(-E(x, y))$
- Synthesis of samples from $p(x|y)$ via Langevin dynamics on E(x,y).
- ( Keep optimization researchers busy ;-) )

# Energy-based models in computer vision and machine learning

Computer vision:

- Discrete optimization based on MAP-inference for Markov random fields (MRFs, CRFs) [Blake, Kohli, Rother '11]
- Continuous optimization for variational models [Mumford, Shah '89]
- Different hybrid forms such as continuous valued MRFs or minimal partitions [Chambolle, Cremers, P. '12]

Machine learning:

- Energy-based model based on neural networks have a long tradition[Hinton et al '03], [LeCun et al. '06], [Du, Mordatch '19], ...
- The discriminator in a Wasserstein GAN can also be seen as some sort of energy-based model [Arjovsky et al. '17]
- Any deep-learning based classifiers (with a sofmax tail) can be related to energy-based models and opens up generative learning [Grathwohl 2020]

# How to train energy-based models?

Assume we have given a set of ground-truth output-input pairs $(x_n, y_n)$. Let us consider an energy $E_\theta(x, y)$ parametrized by some parameter vector $\theta \in \Theta$.

- **Contrastive learning:** [Hinton '02] Find $\theta$ such that $E_\theta(x, y)$ has a low energy on ground truth pairs $(x_n, y_n)$ and high energy on contrastive pairs $(\tilde{x}, y_n)$:

$$\min_\theta \sum_n E_\theta(x_n, y_n) - \sum_n E_\theta(\tilde{x}_n, y_n))$$

- **Bilevel optimization:** [Samuel, Tappen '09] Find $\theta$ such that the solutions $\hat{x}_n \in \operatorname{argmin}_x E(x, y_n)$ minimize a loss function $\mathcal{L}(x_n, \hat{x}_n)$:

$$\min_\theta \sum_n \mathcal{L}(x_n, \hat{x}_n), \quad \text{s.t. } \hat{x}_n \in \operatorname*{argmin}_x E_\theta(x, y_n)$$

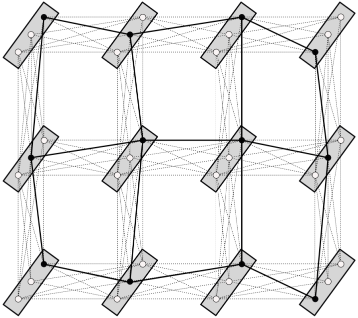
- **Unrolling:** [Domke '12] Find $\theta$ such that the $K$-th iterate $x_n^K$ of an iterative algorithm $\mathcal{A}$ minimizes a loss function $\mathcal{L}(x_n, x_n^K)$:

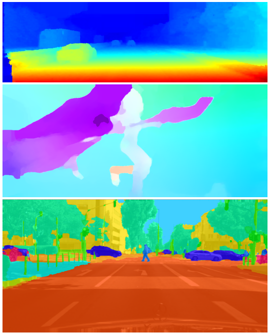
$$\min_\theta \sum_n \mathcal{L}(x_n, x_n^K), \quad \text{s.t. } x_n^{k+1} = \mathcal{A}(E_\theta, x_n^k),\ k = 0, \ldots, K-1$$

- **Black-box differentiation:** [Vlastelica et al. '19]. Restricted to linear objective functions.

# Image labeling / MAP inference

- Many problems in image processing /computer vision can be cast as graph labeling problems [Savchynskyy '19]
- Nodes $i \in \mathcal{V}$ correspond image pixels
- Edges $(i, j) \in \mathcal{E}$ define a neighborhood system
- Each node $i$ can take a label $y_i \in \mathcal{Y} = \{1, 2, \ldots, K\}$
- Assign an energy to a certain configuration of the labels $y = (y_i)_{i \in \mathcal{V}}$.
- Task: Find the configuration with the lowest energy





Stereo

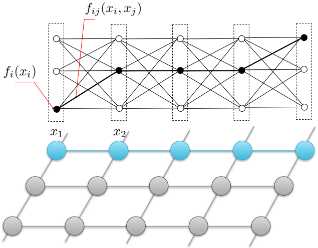
Motion

Segmentation

# The energy

- Discrete optimization models can efficiently impose a smoothness prior.
- We consider the following classical image labeling model:

$$\min_{y \in \mathcal{Y}^{|\mathcal{V}|}} \left\{ E(y|\theta) := \sum_{i \in \mathcal{V}} \theta_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(y_i, y_j) \right\}.$$




- The model depends on unary terms $\theta_i$ as well as binary terms $\theta_{i,j}$.
- Minimizing along a certain chain amounts for solving a shortest path problem.

# Markov random fields

- The energy of the image labeling problems can be interpreted as a negative log posterior distribution

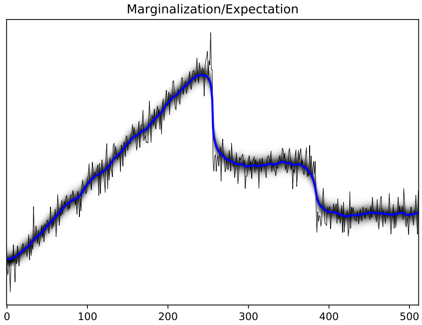
$$p(y|\theta) = \frac{1}{Z}\exp\left(-\frac{E(y|\theta)}{T}\right) = \frac{1}{Z}\prod_{i=1}^{n}\underbrace{\exp(-\theta_i(y_i)/T)}_{\phi_i(y_i)}\prod_{i=1}^{n-1}\underbrace{\exp(-\theta_{i,i+1}(y_i,y_{i+1})/T)}_{\psi_{i,i+1}(y_i,y_{i+1})},$$

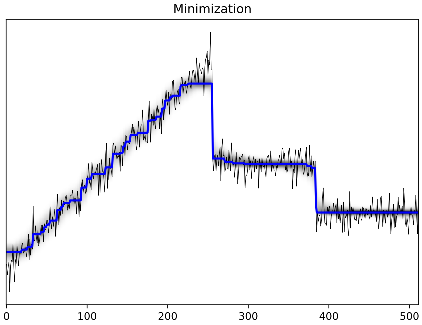
  where $Z$ is the partition function and $T > 0$ is a variance parameter.
- **MAP inference** maximizes $p(y|\theta)$ or minimizes $E(y|\theta)$.
- **Marginalization** requires to compute the marginals distributions $p(y_i|\theta)$,

$$p(y_i|\theta) = \sum_{y_1}\sum_{y_2}\dots\sum_{y_{i-1}}\sum_{y_{i+1}}\dots\sum_{y_n}p(y|\theta).$$

- Both problems can be computed on tree-like graphs using dynamic programming.

# Marginalization vs. Minimization




- The energy is given by quadratic unaries and total variation (TV) pairwise terms.
- The variance parameter was set to $T = 0.02$ (no effect on the minimization).
- The minimization leads to the well-known staircasing artifacts of TV.
- The marginalization followed by computing the expectation seems more natural.

# Lifting

- The image labeling problem can be re-cast as a binary optimization problem by means of lifting.
- Lift labels $y_i$ to vectors $x_i = \mathbf{1}_{y_i}$.
- For example for $K = 5$, $y_i = 3$, one has $x_i = (0, 0, 1, 0, 0)$
- The image labeling problem becomes

$$\min_x \left\{ E(x, f) := \sum_{i \in \mathcal{V}} \theta_i^T x_i + \sum_{(i,j) \in \mathcal{E}} \mathrm{tr}(\theta_{i,j}^T (x_i \otimes x_j)) \right\}.$$

- Has close relations to functional lifting approaches such as [Alberti, Bouchitte, Dal Maso '03] for the Mumford-Shah functional.

# Schlesinger's LP relaxation

- Replace binary variables $x_i$ and $x_i \otimes x_j$ by $v_i$ and $w_{i,j}$.
- Leads to the classical LP relaxation due to [Schlesinger '76]

$$\begin{aligned} \min_{v,w} \quad & \sum_{i \in \mathcal{V}} \theta_i^T v_i + \sum_{(i,j) \in \mathcal{E}} \mathrm{tr}(\theta_{i,j}^T w_{i,j}), \\ \text{s.t.} \quad & v_i^T \mathbf{1} = 1, \quad v_i^l \geq 0, \\ & w_{i,j}^T \mathbf{1} = v_j,\ w_{i,j} \mathbf{1} = v_i, \quad w_{i,j} \geq 0. \end{aligned}$$

- Known as the marginal polytope relaxation.
- Gives favorable integrality gap guarantees [Chekuri et al '04].
- Can be cast as a huge $\mathcal{O}(N^2K^2)$ LP in standard form
- Example: Image size $N \times N = 1024^2$, $K = 256$ labels, $\mathcal{O}(N^2K^2) \sim 7 \cdot 10^{10}$ variables.
- Important observation: The dual problem is much smaller, only $\mathcal{O}(N^2K) \sim 3 \cdot 10^8$ variables.

## Some state-of-the-art algorithms

- Solving the marginal polytope relaxation using an off-the-shelve LP solver is very slow.
- **Max product belief propagation (BP) [Pearl '88] is exact on trees. For graphs with loops it may not converge.**
- Move making algorithms based on graph cuts [Boykov, Veksler, Zabih '01], [Komodakis, Tziritas '07].
- Tree reweighted message passing (TRW) [Wainwright, Jaakkola, Willsky '05] decomposes graph into trees. BP is used as a subroutine on the trees.
- TRW-S [Kolmogorov '06] is a sequential version of TRW which yields a monotonous coordinate ascent on the dual.
- Smoothing approach ( $\min(\cdot) \rightsquigarrow -\mathrm{logsumexp}(-\cdot)$) and Nesterov's accelerated gradient descent [Savchynskyy et al.'11]
- **Dual minorize maximize (DMM), highly parallel monotonous block coordinate ascent [Shekhovtsov et al. '16].**
- **Saddle-point algorithms featuring linearly convergent Frank-Wolfe algorithms [Kolmogorov, P. '21]**

# Solving labeling problems on a chain

- Let us restrict our image labeling problem to one line (chain) of the image:
- On this chain, we consider a graph of $n$ nodes $\mathcal{V} = \{1, 2, \ldots, n\}$, representing the image pixels.
- The edge set is given by pairs of neighboring nodes along that line, that is $\mathcal{E} = \{(i, i+1) : \ i = 1, \ldots, n-1\}$.
- Each node $i \in \mathcal{V}$ can take a label out of a given label set $\mathcal{Y}$.

$$\min_{y_1, \ldots, y_n} E(y_1, \ldots, y_n) := \sum_{i=1}^{n} \theta_i(y_i) + \sum_{i=1}^{n-1} \theta_{i,i+1}(y_i, y_{i+1}).$$

# Main observation

- The pairwise structure of the the energy admits the following recursive definition.

$$E(y_1, \ldots, y_n) = \min_{y_1,\ldots,y_n} \sum_{i=1}^{n-1} \theta_i(y_i) + \sum_{i=1}^{n-1} \theta_{i,i+1}(y_i, y_{i+1}) + \theta_n(y_n)$$

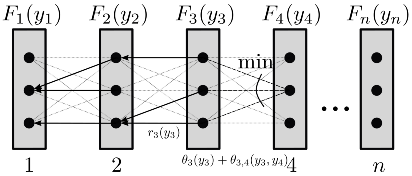
$$= \min_{y_2,\ldots,y_n} \underbrace{\min_{y_1} \theta_1(y_1) + \theta_{1,2}(y_1, y_2)}_{F_2(y_2)} + \sum_{i=2}^{n-1} \theta_i(y_i) + \sum_{i=2}^{n-1} \theta_{i,i+1}(y_i, y_{i+1}) + \theta_n(y_n)$$

$$= \min_{y_3,\ldots,y_n} \underbrace{\min_{y_2} F_2(y_2) + \theta_2(y_2) + \theta_{2,3}(y_2, y_3)}_{F_3(y_3)} + \sum_{i=3}^{n-1} \theta_i(y_i) + \sum_{i=3}^{n-1} \theta_{i,i+1}(y_i, y_{i+1}) + \theta_n(y_n)$$

$$= \ldots = \min_{y_n} F_n(y_n) + \theta_n(y_n).$$

# Dynamic Programming

- The recursive definition allows for an efficient dynamic programming scheme




Source: [Savchynskyy '19]

- In the computation of $F_n$, it turns out that it is convenient to define the functions $F_i : \mathcal{Y} \to \mathbb{R}$:

$$F_1(s) = 0, \quad F_i(s) := \min_{t \in \mathcal{Y}} F_{i-1}(t) + \theta_{i-1}(t) + \theta_{i-1,i}(t, s),$$

which are the so-called Bellman functions or forward messages.

- The same algorithm can be run backwards, the backward messages are then computed as

$$B_n(s) = 0, \quad B_i(s) = \min_{t \in \mathcal{Y}} B_{i+1}(t) + \theta_{i+1}(t) + \theta_{i,i+1}(s, t)$$

# Min-marginals

- The min-marginals at a node $w \in \{1, \dots, n\}$ are defined as the energy $E_w$ at the node $w$ which is obtained by "minimizing out" all other nodes, that is

$$E_w(s) = \min_{y \in \mathcal{Y}^{|\mathcal{V}|}, y_w = s} E(y_1, \dots, y_n; \theta)$$

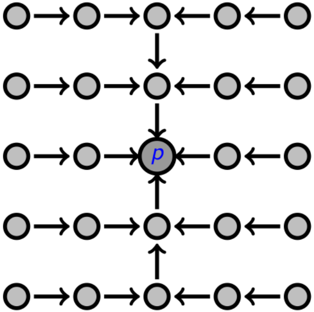
- This means that we are fixing the label of $y_w = s$ but choose all other labels such that they minimize the overall energy.
- Using the definitions of forward messages $F_i$ and backward messages $B_i$ one can easily see that the min-marginals are computed as

$$E_i(s) = F_i(s) + B_i(s) + \theta_i(s).$$

- After computing all min-marginals $E_i(s)$, the optimal labels can be simply computed by picking in each node that label with the smallest energy.

# Extension to grid-like graphs

- Unfortunately, the dynamic programming algorithms cannot be directly extended to grid-like graphs such as images
- However, one can consider iterative algorithms such as loopy belief propagation (BP) that reuses messages from previous iterations.
- A very efficient variant, called sweep BP alternates the largest trees through a particular node [Tappen, Freeman, '03].



# Sweep BP

- Initialize all messages $F_i^h(s), B_i^h(s), F_i^v(s), B_i^v(s)$ with zeros

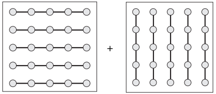
1. for $it = 1, 2, \ldots$ do
2. Compute $F_i^h(s)$, $B_i^h(s)$ with unaries $\tilde{\theta}_i(s) = \theta_i(s) + F_i^v(s) + B_i^v(s)$.
3. Compute $F_i^v(s)$, $B_i^v(s)$ with unaries $\tilde{\theta}_i(s) = \theta_i(s) + F_i^h(s) + B_i^h(s)$.
4. end for
5. Compute min-marginals $E_i(s) = F_i^h(s) + B_i^h(s) + F_i^v(s) + B_i^v(s) + \theta_i(s)$.
6. Select optimal label: $y_i = \arg\min_s E_i(s)$.

- The algorithm does not give any guarantees for convergence, but works very well in practice.
- There exist several variants, which fix this theoretical shortcoming, e.g. TRW-S.

# Dual decomposition

- The labeling energy (primal problem) can be naturally split into problems acting on horizontal and vertical chains

$$\min_y E(y) := E_1(y) + E_2(y),$$



- Problem is easy to minimize in $E_1$ or $E_2$ but not jointly
- Consider the following splitting:

$$\min_y E_1(y) + E_2(y) = \min_{y_1, y_2} E_1(y_1) + E_2(y_2), \text{ s.t. } y_1 = y_2$$

- The corresponding Lagrange dual is given by

$$\begin{aligned} D(\lambda) &= \min_{y_1, y_2} E_1(y_1) + E_2(y_2) + \lambda^T(y_1 - y_2) \\ &= -\left(E_1^*(-\lambda) + E_2^*(\lambda)\right) \end{aligned}$$

- Of course, weak duality holds

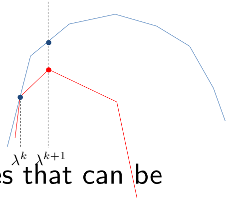
$$\max_\lambda D(\lambda) = \min_y E_1^{**}(y) + E_2^{**}(y) \le \min_y E(y)$$

- This relaxation is equivalent to the marginal polytope relaxation.

# Dual minorize-maximize

- In [Shekhovtsov et al. '16] we have a dual minorize-maximize (DMM) algorithm of the form

$$\begin{cases} \lambda^{k+\frac{1}{2}} = \max_{\lambda} \underline{D}^{1,k}(\lambda) + D^2(\lambda) \\ \lambda^{k+1} = \max_{\lambda} D^1(\lambda) + \underline{D}^{2,k}(\lambda), \end{cases}$$




- A minorant $\underline{D}^{1,k}(\lambda)$ is given by the maximum amount of unaries that can be removed from $D^1(\lambda^k)$ without changing its minimizer
- Yields a monotonically increasing sequence of dual function values
- A highly practical maximal modular minorant is obtained from a hierarchical dynamic programming approach
- Can be efficiently parallelized on the GPU
- Works very well in practice.
- **Disadvantage:** The algorithm might get stuck in non-optimal points.

# A more general optimization problem

- Let us consider a more general discrete optimization problem
$$\min_{X \in \{0,1\}^d} \sum_{t \in T} f_t(X_{A_t}),$$
where we assume that the subproblems are tractable (e.g. DP on chains)
- In particular, we assume, we have an efficient linear minimization oracle `lmo` that can solve for a given $Y$ the problem $\min_{X \in \{0,1\}^{A_t}} f_t(X) + \langle X, Y \rangle$ .
- This problem can be written equivalently as
$$\min_{X \in \{0,1\}^d, X^1 \in \mathcal{P}^1, \ldots, X^m \in \mathcal{P}^m} \sum_{t \in T} X_\circ^t \qquad \text{s.t.} \quad X_v = X_v^t \qquad \forall t \in T, v \in A_t,$$
with polytopes $\mathcal{P}^t = \mathtt{conv}(\{[X \ f(X)] \mid X \in \operatorname{dom} f_t\}) \subseteq \mathbb{R}^{A_t} \times \mathbb{R}$ and $X_\circ^t$ denotes the last component of vector $X^t \in \mathbb{R}^{A_t} \times \mathbb{R}$
- After dropping the $X \in \{0,1\}^d$ constraint, this can be written as the following convex-concave saddle-point problem
$$\min_{x=(X^1,\ldots,X^m) \in \mathcal{P}^1 \times \ldots \times \mathcal{P}^m} \ \max_{y=(Y_v^t)_{t \in T, v \in A_t} \in \mathcal{Y}} \sum_{t \in T} X_\circ^t + \sum_{t \in T, v \in A_t} X_v^t Y_v^t, \quad \text{s.t.} \sum_{t : v \in A_t} Y_v^t = 0$$

# One-sided Frank-Wolfe algorithms for saddle problems *

The previous problem is just an instance of saddle-point problems of the form

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \mathcal{L}(x, y) := \left\langle Kx,\, y \right\rangle + f_{\mathcal{P}}(x) - h^*(y), \quad F(x) = \max_{y \in \mathcal{Y}} \mathcal{L}(x, y), \quad H(y) = \min_{x \in \mathcal{X}} \mathcal{L}(x, y).$$

- The function $f_{\mathcal{P}} = f(x) + \delta_{\mathcal{P}}(x)$ is the sum of a smooth function with Lipschitz continuous gradient and the indicator of a convex polytope and we assume the existence of an efficient **linear minimization oracle** ($\texttt{lmo}$)

$$\texttt{lmo}_{\mathcal{P}}(a) \in \arg\min_{x \in \mathcal{P}} \langle a,\, x \rangle \,.$$

- The function $h^*$ is a convex function which allows to efficiently compute its **proximal map** $(\mathrm{prox})$

$$\mathrm{prox}_{\tau h^*}(\bar{y}) = \arg\min_{y \in \mathcal{Y}} \frac{1}{2\tau} \left\| y - \bar{y} \right\|^2 + h^*(y).$$

*Joint work with V. Kolmogorov, ICML 2021

# Accelerated dual proximal point algorithm

- In case $f(x)$ is a quadratic function and $h^*(y)$ is a linear constraint we can consider a proximal regularization on the dual:

$$\mathcal{L}_{\gamma,\bar{y}}(x,y) = \mathcal{L}(x,y) - \frac{1}{2\gamma}\left\|y-\bar{y}\right\|^2, \quad F_{\gamma,\bar{y}}(x) := \max_{y\in\mathcal{Y}} \mathcal{L}_{\gamma,\bar{y}}(x,y), \quad H_{\gamma,\bar{y}}(y) := \min_{x\in\mathcal{X}} \mathcal{L}_{\gamma,\bar{y}}(x,y).$$

- The iterations of an inexact dual proximal point algorithm are given by

$$(\hat{x},\hat{y}) \approx_{\varepsilon} \operatorname*{argmin}_{(x,y)\in\mathcal{X}\times\mathcal{Y}} F_{\gamma,\bar{y}}(x) - H_{\gamma,\bar{y}}(y),$$

- In $x$, it is the minimization of a quadratic function over a polytope, which can be solved using linearly convergent Frank-Wolfe algorithms: AFW [Lacoste-Julien, Jaggi '15], [Beck, Shtern '17], BCG [Braun at al. '19], DiCG [Garber, Meshi '16] .
- In $y$ is is just the evaluation of the proximal map of $h^*$.
- Allows the application of an inexact accelerated proximal-point algorithm [Aujol, Dossal, '15] where we prescribe the solution accuracy of the problem in $x$.

# Convergence rate

**Theorem.** Assume the (negative) dual problem $-H(y)$ is coercive, and the subproblems in $x$ are solved with a linearly convergent FW method, then the inexact accelerated proximal point algorithm makes $O(n \log n)$ calls to `lmo` during the first $n$ iterations, and the dual iterations satisfy:

$$H(y^*) - H(y_n^e) = O(1/n^2).$$

- Can solve the relaxation exactly with guaranteed convergence rate.
- But still not as fast as BP or DMM.

# Primal-dual algorithm

- In case $f$ and $h$ are more general, we can still apply the inexact primal-dual algorithm [Rasch, Chambolle, '20].
- The proximal subproblems are given by

$$\mathtt{prox}_{\tau f_{\mathscr{P}}}(\bar{x}) = \arg\min_{x \in \mathscr{P}} f(x) + \frac{1}{2\tau} \|x - \bar{x}\|^2 .$$

- Can by again solved approximately using a linearly convergent FW algorithm.

**Theorem.** The inexact primal-dual algorithm makes $O(n \log n)$ calls to `lmo` during the first $n$ iterations for which the dual iterates satisfy

$$H(y^*) - H(y_n^e) = O(1/n).$$

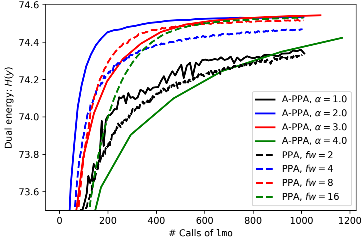
Moreover, if $\mathrm{dom} h^*$ is a compact set, the primal iterates also satisfy

$$F(x_n^e) - F(x^*) = O(1/n).$$

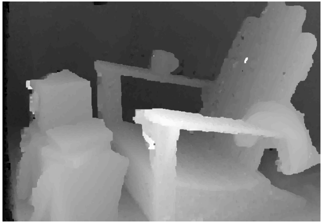
# Some results

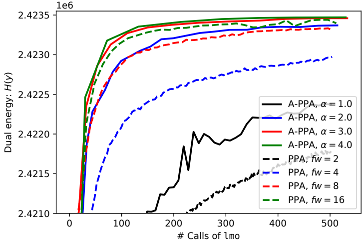
(a) Noisy image

(b) Denoised image

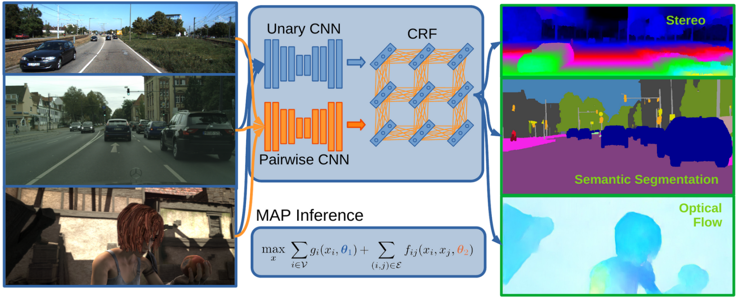



(c) Convergence





# Learning




- For decades, the unary and pairwise terms have been computed based on hand-crafted features
- By the recent advanced of deep learning, it is more than natural to compute unary and pairwise terms based on deep convolutional networks (CNNs)

# Method I: Surrogate loss*

- Ideally, we would like to solve the bilevel optimization problem:

$$\min_\theta \mathscr{L}(x_n, \hat{x}_n), \quad \hat{x}_n \in \operatorname*{argmin}_x E_\theta(x, y_n)$$

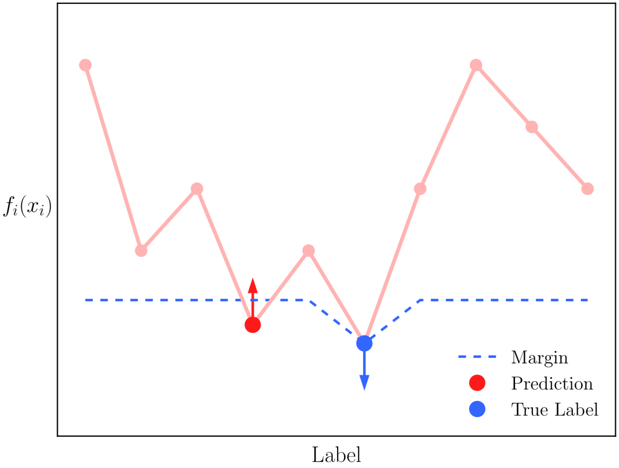
- We can construct an upper bound based on the margin rescaling technique [Tsochantaridis et al. '04]:

$$\begin{aligned}\max_{x \in \operatorname{argmin}_x E_\theta(x,y_n)} \mathscr{L}(x_n, x) &\leq \max_{x : E_\theta(x,y_n) \leq E_\theta(x_n,y_n)} \mathscr{L}(x_n, x) \\ &\leq \max_{x : E_\theta(x,y_n) \leq E_\theta(x_n,y_n)} \mathscr{L}(x_n, x) + E_\theta(x_n, y_n) - E_\theta(x, y_n) \\ &\leq \max_x \mathscr{L}(x_n, x) + E_\theta(x_n, y_n) - E_\theta(x, y_n) \\ &= E_\theta(x_n, y_n) - \min_x E_\theta(x, y_n) - \mathscr{L}(x_n, x)\end{aligned}$$

- Computing the upper bound requires to call the CRF solver.
- The solver can be treated as a black-box.

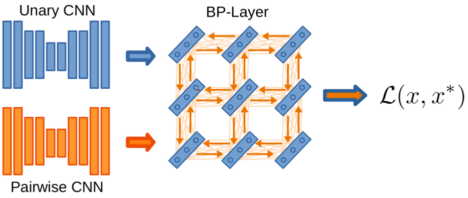
* Joint work with P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, CVPR 2017

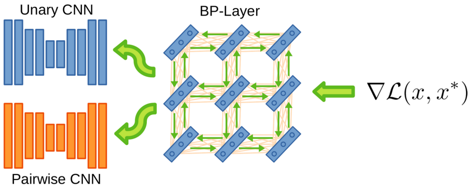
# Graphical explanation




Example using a loss of the form $\mathscr{L}(x, g) = \min\{\tau, |x - g|\}$

# Method II: Unrolling of Loopy belief propagation*




- The normalized min-marginals $\tilde{E}_i(s)$ computed by the loopy BP provide a good approximation to the true marginals.

$$\tilde{E}_i(s) = \frac{\exp(E_i(s))}{\sum_t \exp(E_i(t))}$$

- For learning, we unroll a few iterations of BP and use a cross-entropy loss function defined on the min-marginals.
- It turns out that the derivatives of each BP iterations can be computed efficiently using again dynamic programming on chains.
- The solver must be treated as a white-box.

* Joint work with P. Knöbelreiter, C. Sormann, F. Fraundorfer, A. Shekhovtsov, CVPR 2020

# Some results

Stereo

Motion

# Conclusion/Discussion

- Energy (=optimization) based models for computer vision
- Fast solvers are important for learning and inference
- There is still a performance gap between fast heuristical methods and methods with guaranteed convergence rate.
- Discussed methods for learning that can deal with the combinatorial nature of the models (Black-box vs. white-box).

# Acknowledgements

Thank you for listening!

European Research Council
Established by the European Commission

Der Wissenschaftsfonds.

https://www.tugraz.at/institute/icg/research/team-pock