



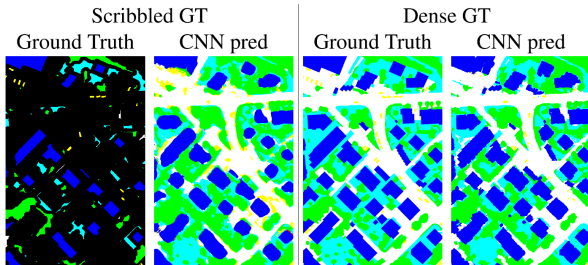
From MAP to MPM: Quantum Annealing for Sampling-Based Inference in CRFs for Semantic Segmentation with Sparse Ground Truth

Daniele Franch

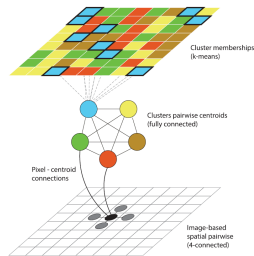
Electrical and Computer Engineering

PhD Retreat

October 13, 2026



- ▶ **Dense GT is expensive:** only scribbles ($\approx 20\%$ of pixels, no boundaries, objects missing).
- ▶ A scribble-trained CNN **do not perform well**
- ▶ **Their fix:** CNN + CRF
 k -means clusters of CNN activations for **long-range context**.

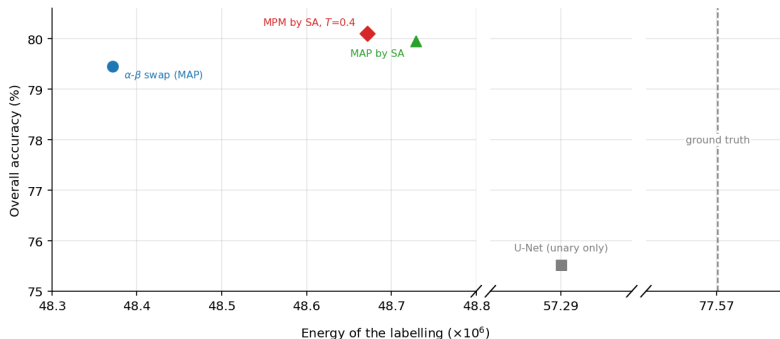


Energy (labels y of pixels i and clusters c):

$$\begin{aligned} E_{CRF}(y) = & \sum_i D_i(y_i) + \lambda_{II} \sum_{i \sim j} V(y_i, y_j) \\ & + \gamma \sum_c D_c(y_c) + \lambda_{CC} \sum_{c \neq d} V(y_c, y_d) \\ & + \lambda_{IC} \sum_i \sum_{c \in \mathcal{N}_h(i)} V(y_i, y_c) \end{aligned}$$

Inference: MAP by α - β swap.

First SA runs: the lowest energy is not the best labelling



- **Energy order** $E(\alpha\beta) < E(\text{MPM}) < E(\text{MAP-SA}) < E(\text{U-Net}) \ll E(\text{GT})$,
but **OA order** $\text{MPM} > \text{MAP-SA} > \alpha\beta > \text{U-Net}$.

The minimum is not the target: GT has 60% more energy than the minimum, 80% of it from the unary term (scribble-trained U-Net).

MAP (α - β swap)

$$\hat{x} = \arg \min_x E(x; y)$$

MPM

$$\hat{x}_i = \arg \max_k p_T(x_i=k | y)$$

$T \rightarrow 0$: MAP, $T \rightarrow \infty$: U-Net

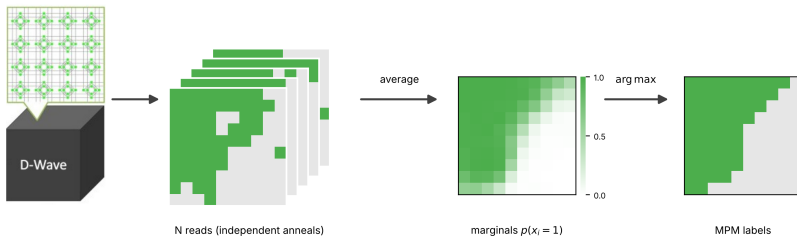
Why MPM

1. **No need for a stopping rule**
2. **Flat energy where the uncertainty lies:**
scribble-trained unaries $\rightarrow$ MPM takes the label
most low-energy maps agree on
3. **Clusters:** MAP commits to one label per cluster,
MPM averages over them

Where the gain comes from

- ▶ **Not reaching the minimum:** most of the gain is shared with MAP by SA, which gains only by stopping early
- ▶ **Pixels move, clusters follow:** 1.3–1.5% of clusters change label; their pixels (7%) carry $\sim$ all the gain
- ▶ **MPM resists the cluster's pull**
- ▶ **More objects recovered:** vs α - β swap, +29 objects found, -20 lost (net +9)

The quantum annealer as an MPM sampler



$$\text{Reads} \approx p(\mathbf{x}) \propto e^{-\beta_{\text{phys}} k E(\mathbf{x})}, \quad T_{\text{eff}} = 1/(\beta_{\text{phys}} k)$$

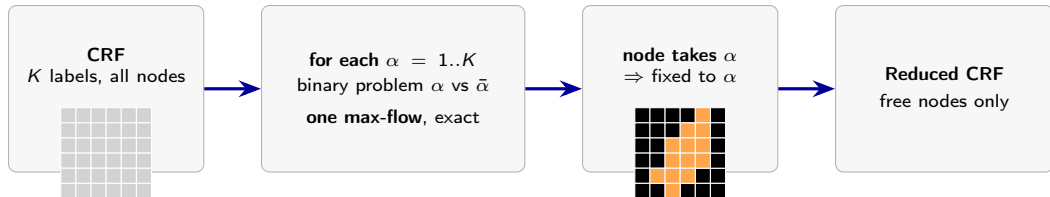
Pros

- ▶ No starting state: no burn-in
- ▶ Each read is a new anneal: no correlation
- ▶ Fast: μs anneals, hundreds of reads per call

Cons

- ▶ Independent but biased: freeze-out and programming errors do not average out
- ▶ T not a direct parameter
- ▶ **Binary only, limited size and connectivity:**
Advantage2 $\approx 4,400$ qubits

Step 1 – Kovtun partial optimality: fewer variables



Pros

- ▶ **Faster**
- ▶ $\simeq$ **Same accuracy**
- ▶ Free set = uncertain pixels: the sampler works only where MPM can matter

Cons

- ▶ **No gain on fixed nodes:** they keep the MAP label, as α - β swap
- ▶ Fragmented free set $\rightarrow$ cut-and-pack

Step 2 – Top-2 QUBO: from $K = 5$ bits to 1 bit per pixel



One-hot: K bits x_{vk} per pixel

$$E = \sum_{v,k} U_v(k) x_{vk} - \sum_{(v,w)} P_{vw} \sum_k x_{vk} x_{wk} + \underbrace{A \sum_v \left(\sum_k x_{vk} - 1 \right)^2}_{\text{one-hot penalty}}$$

- keep the 2 best labels (A_v, B_v)
- no penalty, no invalid reads
- embeddable: $\simeq 200 \rightarrow >2000$ pixels
- **cost: only 2 labels**

Top-2: 1 bit x_v per pixel

$$E = \sum_v \Delta_v x_v + \sum_{(v,w)} P_{vw} [\ell_v(x_v) \neq \ell_w(x_w)]$$

$x_v=0 \rightarrow A_v, x_v=1 \rightarrow B_v, \quad \Delta_v = U_v(B_v) - U_v(A_v)$

How often is the label lost?

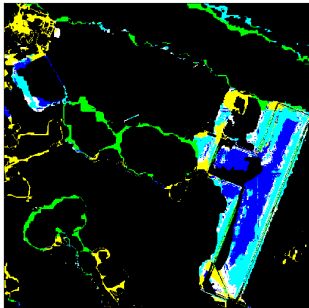
Entropy	not in top-2 (%)	
	true	$\alpha\beta$ -swap
[0.0, 0.1)	2.91	0.00
[0.1, 0.2)	7.69	0.03
[0.2, 0.3)	11.54	0.46
[0.3, 0.4)	13.71	1.55
[0.4, 0.5)	14.56	2.86
[0.5, 0.6)	15.55	4.35
[0.6, 0.7)	16.71	6.33
[0.7, 0.8)	19.12	9.07
[0.8, 0.9)	24.78	15.06
[0.9, 1.0)	36.23	30.61
all	13.26	6.23

$\alpha\beta$ swap error: **20.55%**

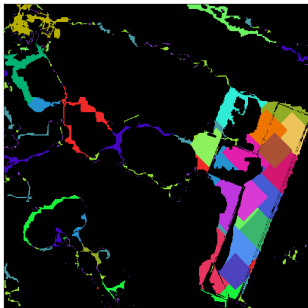
After partial optimality: how to send the free pixels to the QPU?



Free pixels (~ 50000 pixels)



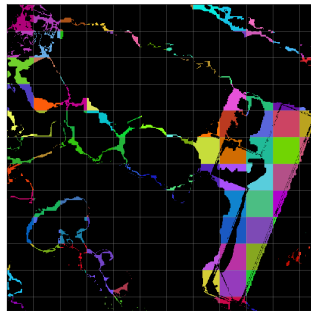
A. Blocks (~ 2500 – 2900 var)



✓ fewest calls

✗ new embedding per block

B. Fixed windows



✓ one embedding for all calls

✗ mostly empty: too many calls

Goal: one embedding (B) + few submissions (A) $\Rightarrow$ **cut-and-pack**

1. **Split** the free pixels into connected components
2. **Cut** every component larger than 52×52 :
 - ▶ at the **narrowest neck**: the position crossed by the fewest pixel edges
 - ▶ repeat until every piece fits in 52×52
3. **Pack** the pieces into the grids:
 - ▶ largest pieces first; try the open grids in order
 - ▶ Check the feasible positions with FFT
 - ▶ keep the position with the most contact
 - ▶ no grid fits → open a new grid
4. **Sample** grid by grid, the rest frozen: **2 passes**, the first is burn-in (discarded), marginals from the second

1. Free pixels after Kovtun



Accuracy results (4 tiles)



Method	OA	mIoU	energy ($\times 10^6$)	subm.	T_{eff} (QPU)
<i>Full label set (5 seeds)</i>					
MPM SA $T=0.4$	80.09%	61.11%	48.67	—	—
Partial optimality + MPM SA $T=0.4$	80.02%	60.93%	48.62	—	—
MAP SA	79.95%	60.95%	48.73	—	—
α - β swap	79.45%	60.39%	48.37	—	—
U-Net (unary only)	75.51%	54.26%	57.29	—	—
<i>Partial optimality + top-2 (1 seed)</i>					
MPM SA $T=0.4$	79.55%	59.80%	49.69	—	—
QPU forward $t_a=5 \mu\text{s}$	79.53%	59.82%	49.66	3942	0.22 ± 0.01
MPM blocks packed $T=0.4$	79.47%	59.67%	49.74	3942	—

- ▶ **MPM > MAP**: +0.64 pp vs α - β swap, despite higher energy
- ▶ **Partial optimality $\approx$ free**: -0.07 pp
- ▶ **Top-2**: -0.47 pp, still > α - β swap in OA

QPU = classical samplers on the reduced problem

Method	Total time		Without network and queue	
	s	s/patch	s	s/patch
<i>Full label set</i>				
MPM SA $T=0.8$	4056	122.9	4056	122.9
α - β swap (MATLAB)	3885	48.6	3885	48.6
<i>Partial optimality + top-2</i>				
MAP ASA	540	6.8	540	6.8
MPM SA $T=0.8$	822	10.3	822	10.3
MPM blocks packed $T=0.8$	3152	39.4	3152	39.4
QPU forward $t_a=5 \mu s$	5562	69.5	1325	16.6

- ▶ **Partial optimality:** MPM becomes cheaper than α - β swap
- ▶ **But** the reduced problem is small: MPM SA remains the fastest MPM
- ▶ **QPU** beats α - β swap if we don't consider network and queue time

As hardware scales, the QPU sampler could overtake the classical ones.

Experiments

- ▶ **Second dataset** (Potsdam)
- ▶ Test different **MPM variants**

Using the marginals

- ▶ **Retrain the U-Net** on the marginals: soft pseudo-labels beyond the scribbles, then iterate

Quantum annealer

- ▶ **Test with different T**
- ▶ **Rerun**: 3 passes (burn-in 0/1/2)



Thank you for your attention!