We are grateful to our three reviewers for their time and insightful comments.

Reviewer 2

“a runtime comparison is needed to understand tradeoffs btw the different choices of projection/decoding sets”

This is a great remark, that we will adress. For decoding, we believe it is important to use the marginal polytope (convex hull) whenever possible, to make sure to predict valid structures. For projections, the knapsack polytope, order simplex and permutahedron have (near) linear-time algorithms so the projection step is very fast. The only potentially problematic case is the Birkhoff polytope, whose projection takes quadratic time. However, we found that the runtimes were surprisingly fast in our experiments. Moreover, quadratic time is still better than the CRF loss, which is intractable. Our provided implementation backs up our claim. We will also report precise runtime figures in the final version.

“Having experiments involving more types of models than just linear [...] could further improve significance”

We focused on linear models to remain in the realm of convex optimization but exploring more models (e.g., neural networks, gradient boosting) would be very interesting indeed. Note however that $\frac{1}{2}\|\varphi(y)-\nabla\Omega^*(\theta)\|^2 \leq \frac{1}{2}\|\varphi(y)-\theta\|^2$ for all $\theta \in \mathbb{R}^d$ and $y \in \mathcal{Y}$ so the Euclidean projection $\nabla\Omega^*(\theta)$ theoretically achieves smaller loss than $\theta = g(x)$ alone, regardless of the model. This is also true for the more general Bregman projection setting.

Reviewer 3

“Limitations include questionable practicality of the approach (due to expensive training and inference) and proximity to previous work [6,26] which is generalized”

Inference / decoding is unavoidable and is used in virtually all structured prediction approaches. The projection step during training takes linear time for the knapsack polytope, order simplex and permutahedron. The only potentially problematic case is the Birkhoff polytope, whose projection takes quadratic time but this is still better than the CRF loss, which is intractable.

Besides consistency analysis, our contributions include the idea that we can use any convex set (and not just the marginal polytope), Prop. 2 which guarantees smoothness of our loss in the KL projection case and Prop. 3 on the order simplex.

“What is the reason for this discrepancy?” (ASGD vs. LBFGS)

ASGD is convenient to analyze when the objective is an expectation (as is the case of the surrogate risk) but in practice we can use any ERM algorithm. We chose LBFGS as it does not require choosing a learning rate.

“Can you comment on calibration of the losses used here w.r.t. the F1 measure?”

For multilabel classification, the unit cube and knapsack polytope lead to losses that are consistent with accuracy (Hamming loss) but there is currently no guarantee for F_1 score. Deriving a polytope and a projection which guarantees calibration w.r.t. F_1 score is actually a great open question we would like to tackle in the future.

Thank you very much for all your other comments, we will address them.

Reviewer 4

“Classical baselines such as SSVM, CRF, SparseMAP etc should be compared.”

In our experiments, we specifically chose tasks for which the CRF loss is not available. For instance, for permutation problems, the CRF loss is intractable, as marginal inference is #P-complete. For multilabel prediction with cardinality constraints, there does not exist any prior algorithm for computing the CRF loss and tractability is an open question.

SparseMAP is already included in our experiments as it corresponds to using Euclidean projections on the marginal polytope (Birkhoff polytope for ranking, knapsack polytope for classification, order simplex for ordinal regression).

A comparison with SSVM would indeed be possible and interesting, although the loss is not consistent and not smooth.

“L100–L106 are key to the paper but are not clearly explained. The technical details should be elaborated [...]

We agree that we were too short on this. We plan to add more details in the final version using the ninth page.

“the influence of different surrogates to the performance is less important than that of the choice of decoding space? E.g., comparing Simplex+B with B+B. Could you make a comment on this?”

It is true that the decoding over $\mathcal{B}$ goes a long way, since it makes sure that a valid permutation matrix is always predicted. But we argue that Proposition 1 is insightful: even using $[0, 1]^k$ instead $\mathbb{R}^k$ improves test accuracy a lot, according to our experiments. Regarding $\mathcal{B}$ vs. $\triangle$, we argue that our empirical results are convincingly in favor of $\mathcal{B}$. Also, when using $\triangle$, we lose the ability to predict soft permutations as shown in Figure 2.