
Deep Reasoning Networks: Thinking Fast and Slow

Di Chen

Cornell University
Ithaca, NY 14853
di@cs.cornell.edu

Yiwei Bai

Cornell University
Ithaca, NY 14853
bywbilly@cs.cornell.edu

Wenting Zhao

Cornell University
Ithaca, NY 14853
wzhao@cs.cornell.edu

Sebastian Ament

Cornell University
Ithaca, NY 14853
ament@cs.cornell.edu

John M. Gregoire

California Institute of Technology
Pasadena CA 91125
gregoire@caltech.edu

Carla P. Gomes

Cornell University
Ithaca, NY 14853
gomes@cs.cornell.edu

Abstract

We introduce Deep Reasoning Networks (DRNets), an end-to-end framework that combines deep learning with reasoning for solving complex tasks, typically in an unsupervised or weakly-supervised setting. DRNets exploit problem structure and prior knowledge by tightly combining logic and constraint reasoning with stochastic-gradient-based neural network optimization. We illustrate the power of DRNets on de-mixing overlapping hand-written Sudokus (Multi-MNIST-Sudoku) and on a substantially more complex task in scientific discovery that concerns inferring crystal structures of materials from X-ray diffraction data under thermodynamic rules (Crystal-Structure-Phase-Mapping). At a high level, DRNets encode a structured latent space of the input data, which is constrained to adhere to prior knowledge by a reasoning module. The structured latent encoding is used by a generative decoder to generate the targeted output. Finally, an overall objective combines responses from the generative decoder (thinking fast) and the reasoning module (thinking slow), which is optimized using constraint-aware stochastic gradient descent. We show how to encode different tasks as DRNets and demonstrate DRNets' effectiveness with detailed experiments: DRNets significantly outperform the state of the art and experts' capabilities on Crystal-Structure-Phase-Mapping, recovering more precise and physically meaningful crystal structures. On Multi-MNIST-Sudoku, DRNets perfectly recovered the mixed Sudokus' digits, with 100% digit accuracy, outperforming the supervised state-of-the-art MNIST de-mixing models. Finally, as a proof of concept, we also show how DRNets can solve standard combinatorial problems – 9-by-9 Sudoku puzzles and Boolean satisfiability problems (SAT), outperforming other specialized deep learning models. DRNets are general and can be adapted and expanded to tackle other tasks.

1 Introduction

Human thought consists of two different types of processes [Kahneman, 2011]: *System 1*, a fast, implicit (automatic), unconscious process, and *System 2*, a slow, explicit (controlled), conscious process. Humans use *System 1* most of the time. *System 1* is fast, effortless, and provides a type of near-automatic pattern recognition. In contrast, *System 2* is slow, rational, requiring more careful thinking, and is used to solve more complex reasoning problems.



Figure 1: (a) Two 4x4 Sudokus: The cells in each row, column, and any of the four 2x2 boxes involving the corner cells have non-repeating digits. (b) Two overlapping Sudokus, with a mixture of two digits in each cell: one from 1 to 4 and the other from 5 to 8. In **Multi-MNIST-Sudoku**, the digits of two overlapping hand written Sudokus (b) have to be de-mixed (as done by DRNet in (c)). (d) The reconstructed overlapping hand written Sudokus from DRNet. (e) A standard **9-by-9 Sudoku puzzle**: a partially filled Sudoku has to be completed as a valid Sudoku.

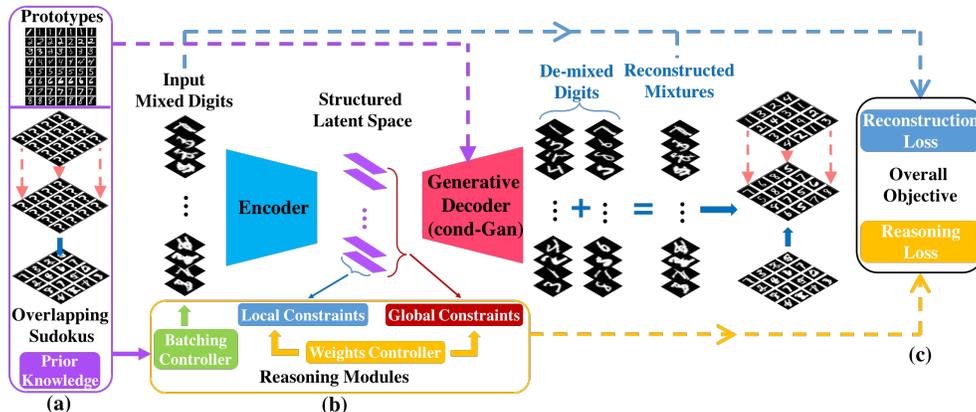


Figure 2: Deep Reasoning Networks (DRNets) perform end-to-end deep reasoning by encoding a latent space of the input data that captures prior knowledge constraints and is used by a generative decoder to generate the desired output. (a) Prior knowledge includes prototypes of digits, which are used to pre-train and build the decoder’s generative module, and Sudoku’s rules, which help DRNet reason about the overlapping digits. (b) Reasoning modules batch data points involved in the same constraints (cells in rows, columns, blocks of a Sudoku) together, enforce that the structure of the latent space satisfies prior knowledge, and dynamically adjust the weights of constraints based on their satisfiability. (c) The overall objective combines responses from the generative decoder (thinking fast) and the reasoning modules (thinking slow).

Deep learning has achieved tremendous success in areas such as vision, speech recognition, language translation, and autonomous driving. Nevertheless, certain limitations of deep learning are generally recognized, in particular, limitations due to the fact that deep learning approaches heavily depend on the availability of large amounts of labeled data. In fact, the current state of the art of deep learning has been compared to *System 1*, i.e., performing pattern recognition or heuristic evaluation. So, when it comes to complex problems that involve reasoning (*System 2*), such as playing Go or crystal structure phase mapping, pure machine learning approaches have to be complemented with reasoning algorithms, such as Monte Carlo tree search [Anthony et al., 2017, Silver et al., 2016, 2018], or mixed-integer programming [Ermon et al., 2015]. Such reasoning approaches are in general outsourced using external modules, which is not always possible and may result in inferior performance due to the coordination barrier between neural networks (*System 1*) and the outsourced reasoning module (*System 2*), which is often non-differentiable. Therefore, an efficient scheme is needed to integrate the two systems *in a general and seamless way*.

We propose **Deep Reasoning Networks (DRNets)**, an end-to-end framework that combines deep learning with logical and constraint reasoning for solving complex tasks that require both *System 1* and *System 2* style thinking, typically in an unsupervised or weakly-supervised setting. We illustrate the power of DRNets for disentangling two overlapping hand-written Sudokus (**Multi-MNIST-Sudoku**) (see Fig.1) and for solving a substantially more complex task in scientific discovery that concerns inferring crystal structures of materials from X-ray diffraction data, which we refer to as **Crystal-Structure-Phase-Mapping**. Both tasks require probabilistic reasoning to interpret noisy and uncertain data, while satisfying a set of rules: Sudoku rules and thermodynamic rules. For example, de-mixing hand written digits is challenging, but it becomes more feasible when we

reason about the prior knowledge concerning the two overlapping Sudokus. Crystal structure phase mapping is yet substantially more complex. In fact, crystal structure phase mapping easily becomes too complex for experts to solve and is a major bottleneck in high-throughput materials discovery. DRNets are motivated and inspired by problems from scientific discovery, such as crystal structure phase mapping.

Our contributions: (1) We introduce **Deep Reasoning Networks (DRNets)**, an end-to-end unsupervised framework that combines deep learning with logical and constraint reasoning. DRNets perform end-to-end deep reasoning by encoding a latent space of the input data that captures the structure and prior knowledge constraints within and among data points (Fig.2). The latent space is used by a generative decoder to generate the desired output, consistent with the input data and prior knowledge. DRNets optimize an objective function capturing the overall problem objective as well as prior knowledge in the form of weighted constraints, using (2) **Constraint-Aware Stochastic Gradient Descent**. DRNets batch data points involved in the same constraint component together and dynamically adjust the constraints' weights as a function of their satisfiability during the optimization phase. (3) We propose a **group of entropy-based continuous relaxations that use probabilistic modelling to encode general discrete constraints including sparsity, cardinality, so-called All-Different constraints, and SAT constraints**. De facto, these examples illustrate how to develop "gadgets" to encode a variety of combinatorial constraints and prior knowledge in DRNets. (4) We show **how to encode Multi-MNIST-Sudoku, standard 9-by-9 Sudoku, SAT, and Crystal-Structure-Phase-Mapping as DRNets**, by properly defining the structure of the latent space, additional reasoning modules to model the problem constraints (prior knowledge), and the components of the objective function. (5) We provide **detailed experimental results** demonstrating the potential of DRNets. In particular, we show how (5.1) DRNets significantly outperformed the state of the art and human experts on **Crystal-Structure-Phase-Mapping instances**, recovering more precise, interpretable, and physically meaningful crystal structure pattern decompositions. (5.2) On **Multi-MNIST-Sudoku instances**, DRNets perfectly recovered the digits in the mixed Sudokus with 100% digit accuracy and outperformed the supervised state-of-the-art MNIST de-mixing models, including CapsuleNet [Sabour et al., 2017] and ResNet [He et al., 2016]. (5.3) DRNets also solve standard **combinatorial problems**, such as **9-by-9 Sudoku puzzles** and **3-SAT** [Mitchell et al., 1992], which require hidden structure reasoning, outperforming the supervised deep-learning state of the art.

While we illustrate the potential of DRNets applied to different variants of Sudoku, 3-SAT problems, and Crystal-Structure-Phase-Mapping, DRNets are general and can be adapted and expanded to many other applications. Future research entails developing the corresponding "gadgets" for incorporating other types of constraints, prior knowledge, and objective functions, for other applications.

2 Related Work

Exploiting problem structure and reasoning about prior knowledge in machine learning tasks has been of increasing interest to facilitate learning, enhance generalization, and improve interpretability [Taskar et al., 2004, Ganchev et al., 2010, Ermon et al., 2015, Hu et al., 2016a]. Bayesian machine learning [Nasrabadi, 2007] imposes prior beliefs by regularizing the posterior with prior distributions. Ganchev et al. [2010] proposed *posterior regularization* (PR), which encodes the soft constraints via a variational distribution. Hu et al. [2016a,b] introduced the PR framework into deep learning for solving natural language processing tasks. In computer vision, symmetry and bone-length constraints were introduced for human pose estimation [Zhou et al., 2017, 2016], and linear constraints were imposed for image segmentation [Pathak et al., 2015]. In structured prediction, Chen et al. [2018] imposed a multivariate Gaussian distribution to capture the correlation among multiple entities, and Lee et al. [2017] incorporate constraints at the inference stage via fine-tuning. In reinforcement learning, Anthony et al. [2017], Silver et al. [2016, 2018] outsource the reasoning process (*System 2*) to external Monte Carlo tree search. In representation learning, k-Sparse autoencoder [Makhzani and Frey, 2013] proposed a k-sparse encoding of the original data. A PCA-like autoencoder [Ladjal et al., 2019] uses a covariance loss term to encourage the dimensions of the latent space to be statistically independent. Deep generative models [Goodfellow et al., 2014, Kingma and Welling, 2013, Oord et al., 2016, Larochelle and Murray, 2011, Hu et al., 2017a] intrinsically impose a prior distribution into the latent space to reason about the original data distribution, which implicitly exploits the underlying structure. InfoGan [Chen et al., 2016] uses mutual information loss to compress most information into an interpretable low-dimensional encoding. Mirza and Osindero [2014], Hu et al.



Figure 3: The reduction flow of Deep Reasoning Networks.

[2017b] use labeled data to control the sample attributes and disentangle the latent space. Hu et al. [2018] introduced *posterior regularization* into deep generative models to learn structured knowledge from labeled data that improves the quality of generated samples.

Leveraging machine learning to solve combinatorial optimization problems has also received much attention (see e.g., Bengio et al. [2018] for a recent survey). For examples: Bello et al. [2016] and Bengio et al. [2018] explored reinforcement learning and Pointer Networks for the *traveling salesman problem*. Li et al. [2018] use graph convolutional networks to guide the local search for solving graph-related NP-complete problems. Selsam et al. [2018], Amizadeh et al. [2019] proposed NeuroSAT and PDP to tackle SAT problems with specialized neural networks and one-bit supervision. Wilder et al. [2018] proposed to use continuous relaxation of discrete problems to backpropagate the gradients, to upstream machine learning models.

While exploiting problem structure and prior knowledge in deep neural networks has received much attention, previous works primarily focus on supervised settings for data-rich domains, typically using large amounts of labeled data, which reduces the importance of explicitly reasoning about prior knowledge given the data’s strong signal. Furthermore, with few exceptions, the constraints proposed in previous models are often independent, soft, i.e., violating them only leads to a worse solution, and are mainly used as regularization terms. In contrast, the problems that DRNets aim to solve often involve many constraints that are hard and correlated: satisfying one constraint while neglecting others can potentially make them unsatisfiable, and violating any of them directly results in an invalid solution (e.g., a wrong Sudoku digit could make the whole puzzle unsolvable). Therefore, a tactical architecture and a smart reasoning module are needed to tackle such challenges. To the best of our knowledge, DRNets are the first end-to-end unsupervised framework that combines deep learning with logical and constraint reasoning for solving complex tasks.

3 Deep Reasoning Networks

DRNets (see Fig.2) are inspired by human thinking [Shivhare and Kumar, 2016]: we abstract patterns to higher-level descriptions and combine them with prior-knowledge to fill-in the gaps. Consider the Multi-MNIST-Sudoku example (Fig.1): we first guess the digits in each cell based on the patterns; we re-adjust our initial beliefs and re-image the overlapping patterns by reasoning about Sudoku rules and comparing them to the original ones, potentially involving several iterations.

Formally, **DRNets formulate unsupervised learning as constrained optimization, incorporating abstractions and reasoning about structure and prior knowledge:**

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(G(\phi_{\theta}(\mathbf{x}_i)), \mathbf{x}_i) \quad \text{s.t. } \phi_{\theta}(\mathbf{x}_i) \in \Omega^{\text{local}} \text{ and } (\phi_{\theta}(\mathbf{x}_1), \dots, \phi_{\theta}(\mathbf{x}_N)) \in \Omega^{\text{global}} \quad (1)$$

In this formulation, $\mathbf{x}_i \in R^n$ is the i -th n -dimensional input data point, $\phi_{\theta}(\cdot)$ is the function of the encoder in DRNets parameterized by θ , $G(\cdot)$ denotes the generative decoder, $\mathcal{L}(\cdot, \cdot)$ is the loss function (e.g., evaluating the reconstruction of patterns), Ω^{local} and Ω^{global} are the constrained spaces w.r.t. a single input data point and several input data points, respectively. $G(\cdot)$ is in general a fixed pre-trained or parametric model. For example, in Multi-MNIST-Sudoku, $G(\cdot)$ is a pre-trained conditional GAN [Mirza and Osindero, 2014] using hand-written digits, and for Crystal-Structure-Phase-Mapping, $G(\cdot)$ is a Gaussian Mixture model. Note that constraints can involve several (potentially all) data points: e.g., in Sudoku, all digits should form a valid Sudoku and in crystal-structure-phase-mapping, all data points in a composition graph should form a valid phase diagram. Thus, we specify local and global constraints in DRNets – local constraints only involve a single input data point whereas global constraints involve several input data points, and they are optimized using different strategies.

Solving the constrained optimization problem (1) directly is extremely challenging since the objective function in general involves deep neural networks, which are highly non-linear and non-convex, and prior knowledge often even involves combinatorial constraints (Fig.3). Therefore, we use Lagrangian

relaxation to approximate equation (1) with an unconstrained optimization problem, i.e.,

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(G(\phi_{\theta}(\mathbf{x}_i)), \mathbf{x}_i) + \lambda^l \psi^l(\phi_{\theta}(\mathbf{x}_i)) + \sum_{j=1}^{N_g} \lambda_j^g \psi_j^g(\{\phi_{\theta}(\mathbf{x}_k) | k \in S_j\}) \quad (2)$$

N is the number of input data points, N_g denotes the number of global constraints, S_j denotes the set of indices w.r.t. the data points involved in the j -th global constraint, and ψ^l, ψ_j^g denote the penalty functions for local constraints and global constraints, respectively, along with their corresponding penalty weights λ^l and λ_j^g . In the following, we propose two mechanisms to tackle the above unconstrained optimization task (Fig.3).

Continuous Relaxation: Prior knowledge often involves combinatorial constraints with discrete variables that are difficult to optimize in an end-to-end manner using gradient-based methods. Therefore, we need to design proper continuous relaxations for discrete constraints to make the overall objective function differentiable. We propose a group of entropy-based continuous relaxations to encode general discrete constraints such as sparsity, cardinality, All-Different constraints, and SAT constraints (see Fig.4). Moreover, our framework can be easily expanded to encode other constraints.

<p>Cardinality Constraint $e_{i,j} \in \{0,1\} \quad j = 1 \dots 8$ s. t. $\sum_{j=1}^4 e_{i,j} = 1$ and $\sum_{j=5}^8 e_{i,j} = 1$</p>	<p>Cardinality Constraint Relaxation $\min_{\theta} H(P_i) + H(Q_i)$ $= -\sum_{j=1}^4 P_{i,j} \log P_{i,j} - \sum_{j=5}^8 Q_{i,j} \log Q_{i,j}$</p>
<p>All-Different Constraint For all constrained set S s. t. $\sum_{i \in S} e_{i,j} = 1$ for $j = 1 \dots 8$</p>	<p>All-Different Constraint Relaxation For all constrained set S $\max_{\theta} H(\bar{P}_S) + H(\bar{Q}_S)$</p>
<p>k-Sparsity Constraint $e_{i,j} \in \{0,1\} \quad j = 1 \dots M$ s. t. $\sum_{j=1}^M e_{i,j} \leq k$</p>	<p>k-Sparsity Constraint Relaxation $\min_{\theta} \max\{H(P), c\}$, where $c < \log k$</p>
<p>Integer Programming Encoding of SAT For any literal x_i and its negation \bar{x}_i, s. t. $x_i, \bar{x}_i \in \{0,1\}$ and $x_i + \bar{x}_i = 1$ For any clause $C_j = a_{j,1} \vee \dots \vee a_{j,K_j}$ $\forall a_{j,k} \in \{x_i, \bar{x}_i\}_{i=1}^n$, s. t. $\sum_{k=1}^{K_j} a_{j,k} \geq 1$</p>	<p>SAT Relaxation For any literal x_i and its negation \bar{x}_i ($i = 1 \dots N_l$), we model a distribution $P_i \sim \text{Bern}(p_i, q_i)$, s. t. $x_i \triangleq p_i$, and $\bar{x}_i \triangleq q_i$ For all clause $C_j = a_{j,1} \vee \dots \vee a_{j,K_j}$, $j = 1 \dots N_c$ $\min_{\theta} -\sum_{j=1}^{N_c} \text{leaky_relu}\left(1 - \sum_{k=1}^{K_j} a_{j,k}\right) + \lambda_h \sum_{j=1}^{N_l} H(P_i)$</p>

Figure 4: Examples of continuous relaxations: $e_{i,j}, N_c, N_l, K_j, \lambda_h, P_i$ denote binary variables, the number of clauses, the number of literals, the number of literals in the j -th clause, the weights of entropy terms, and the Bernoulli distribution for the i -th literal. "leaky_relu" is the leaky ReLU.

We construct continuous relaxations based on probabilistic modelling of discrete variables, where we model a probability distribution over all possible values for each discrete variable. For example, in Multi-MNIST-Sudoku, a way of encoding the possible two digits in the cell indicated by data point x_i (one from $\{1 \dots 4\}$ and the other from $\{5 \dots 8\}$), is to use 8 binary variables $e_{i,j} \in \{0,1\}$, while requiring $\sum_{j=1}^4 e_{i,j} = 1$ and $\sum_{j=5}^8 e_{i,j} = 1$. In DRNets, we model probability distribution P_i and Q_i over digits 1 to 4 and 5 to 8 respectively: $P_{i,j}, j=1 \dots 4$ and $Q_{i,j}, j=1 \dots 4$ denote the probability of digit j and the probability of digit $j+4$, respectively. We approximate the cardinality constraint of $e_{i,j}$ by minimizing the entropy of P_i and Q_i , which encourages P_i and Q_i to collapse to one value. Another combinatorial constraint in Multi-MNIST-Sudoku is the All-Different constraint, where all the cells in a *constrained set* S , i.e., each row, column, and any of four 2x2 boxes involving the corner cells, must be filled with non-repeating digits. For a probabilistic relaxation of the All-Different constraint, we analogously define the entropy of the averaged digit distribution for all cells in a constrained set S , i.e., $H(\bar{P}_S)$:

$$H(\bar{P}_S) = -\sum_{j=1}^4 \bar{P}_{S,j} \log \bar{P}_{S,j} = -\sum_{j=1}^4 \left(\frac{1}{|S|} \sum_{i \in S} P_{i,j} \right) \log \left(\frac{1}{|S|} \sum_{i \in S} P_{i,j} \right) \quad (3)$$

In this equation, a larger value implies that the digits in the cells of S distribute more uniformly. Thus, we can analogously approximate All-Different constraints by maximizing $H(\bar{P}_S)$ and $H(\bar{Q}_S)$. One can see, by minimizing all $H(P_i)$ and $H(Q_i)$ to 0 as well as maximizing all $H(\bar{P}_S)$ and $H(\bar{Q}_S)$ to $\log |S|$, we find a valid solution for the two 4x4 Sudoku puzzles, where all $P_{i,j}$ are either 0 or 1. Furthermore, we can easily generalize those two relaxations for 9x9 Sudoku puzzles.

We also propose to relax the k -sparsity constraints, which for example in Crystal-Structure-Phase-Mapping state the maximum number k of pure phases in an XRD-pattern, by minimizing the entropy of the phase distribution P below a threshold $c < \log k$. We choose the threshold $c < \log k$ because the entropy of a discrete distribution P with at most k positive values cannot exceed $\log k$.

Finally, we approximate the SAT constraints by relaxing their integer programming encoding, where we minimize the entropy of literals to enforce their collapse to either 0 or 1, while maximizing the sum of literals in each clause to encourage one of them to be 1 (true). Moreover, we use "leaky_relu" [Xu et al., 2015] to discourage increasing the sum in each clause when its larger than 1.

Algorithm 1 Constraint-aware stochastic gradient descent optimization of deep reasoning networks.

- Input:** (i) Data points $\{x_i\}_{i=1}^N$. (ii) Constraint graph. (iii) Penalty functions $\psi^l(\cdot)$ and $\psi_j^g(\cdot)$ for the local and the global constraints. (iv) Pre-trained or parametric generative decoder $G(\cdot)$.
- 1: Initialize the penalty weights λ^l, λ_j^g and thresholds for all constraints.
 - 2: **for** number of optimization iterations **do**
 - 3: Batch data points $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ from the sampled (maximal) connected components.
 - 4: Collect the global penalty functions $\{\psi_j^g(\cdot)\}_{j=1}^M$ concerning those data points.
 - 5: Compute the latent space $\{\phi_\theta(\mathbf{x}_1), \dots, \phi_\theta(\mathbf{x}_m)\}$ from the encoder.
 - 6: Adjust the penalty weights λ_l, λ_j^g and thresholds accordingly.
 - 7: minimize $\frac{1}{m} (\sum_{i=1}^m \mathcal{L}(G(\phi_\theta(\mathbf{x}_i)), \mathbf{x}_i) + \lambda_l \psi^l(\phi_\theta(\mathbf{x}_i))) + \sum_{j=1}^M \lambda_j^g \psi_j^g(\{\phi_\theta(\mathbf{x}_k) | k \in S_j\})$ using any standard gradient-based optimization method and update the parameters θ .
 - 8: **end for**
-

Constraint-Aware Stochastic Gradient Descent: We propose constraint-aware SGD to tackle the global penalty functions $\psi_j^g(\{\phi_\theta(\mathbf{x}_k) | k \in S_j\})$, which involve several (potentially all) data points. We define a *constraint graph*, an undirected graph in which each data point forms a vertex and two data points are linked if they are in the same global constraint. Constraint-aware SGD batches data points from the randomly sampled (maximal) connected components in the *constraint graph*, and optimizes the objective function w.r.t. the subset of global constraints concerning those data points and the associated local constraints. For example, in Multi-MNIST-Sudoku, each overlapping Sudoku forms a maximal connected component, we batch the data points from several randomly sampled overlapping Sudokus and optimize the All-Different constraints (global) as well as the cardinality constraints (local) within them. However, in Crystal-Structure-Phase-Mapping, the maximal connected component becomes too large to batch together, due to the constraints (*phase field connectivity* and *Gibbs-alloying rule*) concerning all data points in the composition graph. Thus, we instead only batch a subset (still a connected component) of the maximal connected component – e.g., a path in the composition graph, and optimize the objective function that only concerns constraints within the subset (along the path). By iteratively solving sampled local structures of the "large" maximal component, we cost-efficiently approximate the entire global constraint. Moreover, for optimizing the overall objective, constraint-aware SGD dynamically adjusts the thresholds and the weights of constraints according to their satisfiability, which can involve non-differentiable functions.

For efficiency, DRNets solve all instances together using constraint-aware SGD (see Algorithm 1).

4 Experiments

We illustrate the power of DRNets on two complex tasks and two standard combinatorial problems – disentangling two overlapping hand-written Sudokus (**Multi-MNIST-Sudoku**), inferring crystal structures of materials from X-ray diffraction data (**Crystal-Structure-Phase-Mapping**), solving **9x9 Sudoku Puzzles**, and **3-SAT problems**. We use 3-layer-fully-connected networks as our encoders for all tasks, but we use different generative decoders for different tasks. Moreover, since DRNets are an unsupervised framework, we can apply the *restart* [Gomes et al., 1998] mechanism, i.e., we can re-run DRNets for unsolved instances.

Multi-MNIST-Sudoku: We generated 160,000 input data points which correspond to 32x32 images of overlapping digits coming from the *test set* of MNIST [LeCun et al., 1998] and every 16 data points form a 4-by-4 overlapping Sudokus. Note, our task is more challenging than CapsuleNet’s [Sabour et al., 2017], in which they offset the digits by 4 pixels, while we fully overlap them, explaining

CapsuleNet’s different performance. For Multi-MNIST-Sudoku, the DRNet batches every 16 data points together to enforce the All-Different constraints among the cells of each Sudoku. We use a conditional GAN [Mirza and Osindero, 2014] as our generative decoder (denoted as $G(\cdot)$), which is trained using the digits in the *training set* of MNIST. For each cell x_i , the decoder encodes a latent space, which consists of two parts: The first part includes two distribution P_i and Q_i (see Fig.5) concerning the possible digits in the cell, and the second part is the latent encodings $z_{i,1}, \dots, z_{i,8}$ of each possible digit conditioned on the overlapping digits, which is used by the generative decoder to generate the corresponding digits $G(z_{i,j})$. We obtain our estimation of the two digits in the cell by computing the expected digits over P_i and Q_i , i.e., $\sum_{j=1}^4 P_{i,j} G(z_{i,j})$ and $\sum_{j=1}^4 Q_{i,j} G(z_{i,j+4})$, and reconstruct the original input mixture (see Fig.5). As we described before, we impose the continuous relaxation of cardinality constraints and All-Different constraints to reason about the Sudoku structure among cells of the overlapping Sudokus. To demonstrate the power of reasoning, we compared our unsupervised DRNets with supervised start-of-the-art MNIST de-mixing models – CapsuleNet [Sabour et al., 2017] and ResNet [He et al., 2016], and a variant of DRNets that removes the reasoning modules ("DRNets w/o Reasoning"). We evaluate both the percentage of digits that are correctly de-mixed (digit accuracy) and the percentage of overlapping Sudokus that have all digits correctly de-mixed (Sudoku accuracy). Empowered by reasoning, DRNets significantly outperformed CapsuleNet, ResNet, and DRNets without reasoning, perfectly recovered all digits with the *restart* mechanism (see Table 1), and additionally reconstructed the mixture with high-quality (see Fig.1).

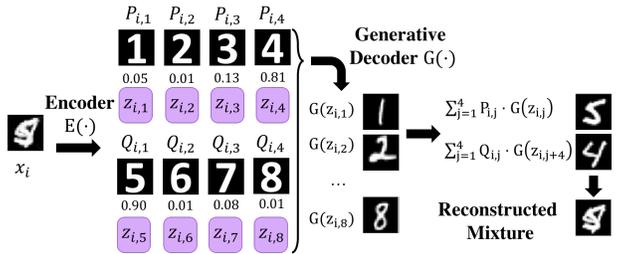


Figure 5: The latent space of the DRNet for Multi-MNIST-Sudoku.

Multi-MNIST-Sudoku (10,000 instances)			
Accuracy (%)	Digit	Sudoku	Time
DRNets + Restart	100.00	100.00	2hours
DRNets	99.99	99.92	2hours
DRNets w/o Reasoning	90.43	20.06	2hours
CapsuleNet	88.46	2.01	2min + 7hrs
ResNet-110	91.44	76.40	5min + 1day

Table 1: Accuracy comparison. We show "test time + training time" for supervised baselines and "solving time" for unsupervised DRNets.

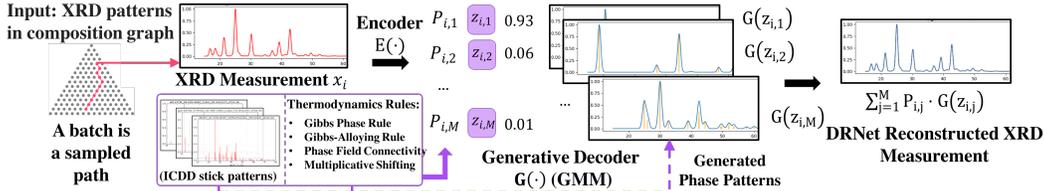


Figure 6: The latent space of the DRNet for Crystal-Structure-Phase-Mapping. M denotes the number of possible phases. (For Al-Li-Fe, $M = 159$; For Bi-Cu-V, $M = 100$.)

Crystal-Structure-Phase-Mapping concerns inferring crystal structures from a set of X-ray diffraction measurements (XRDs) of a given chemical system, given a variety of thermodynamic constraints. Crystal structure phase mapping is a very challenging task: Each X-ray measurement may involve several mixed crystal structures; each chemical system includes hundreds of possible crystal structures; for each crystal structure pattern, we only have a theoretical (idealized) model of pure crystal phases; the rules of thermodynamics are also complex; and the crystal patterns are difficult for human experts to interpret, much more complex than identifying digits. In fact, the current state of the art of crystal structure phase mapping is a major bottleneck in high-throughput materials discovery.

Herein, we illustrate DRNet for crystal structure phase mapping for two chemical systems: (1) a ternary **Al-Li-Fe** oxide system [Le Bras et al., 2014], which is theoretically based, synthetically generated, with ground truth solutions, and (2) a ternary **Bi-Cu-V** oxide system, which is a more challenging real system obtained from chemical experiments and is more noisy and uncertain. For each system, each input data point is the XRD of a mixture of crystal structures. Additionally, the input includes the *composition graph* specifying elemental compositions and the *constraint graph* of the data points. We also collected a library of possible crystal structures from the International Centre for Diffraction Data (ICDD) database. Each crystal structure (also named *phase*) is given

as a list of diffraction peak location-amplitude pairs, (referred to as *stick pattern*), representing the ideal phase patterns measured in a perfect condition (see Fig.6). To model more realistic conditions, DRNets simulate the real phase patterns from *stick patterns* using Gaussian mixture models, where the relative peak locations and mixture coefficients are given by the stick locations and amplitudes. Moreover, the peak width, peak location shift, and peak amplitude variance are parameterized by the latent encoding $z_{i,j}$ and used by the generative decoder to generate the corresponding possible phase patterns in the reconstructed XRD measurement.

We compared DRNets with the state-of-the-art model (IAFD) [Bai et al., 2017], which uses non-negative matrix factorization (NMF), interacting with external mixed-integer programming modules to enforce prior knowledge. For the Al-Li-Fe oxide system, though IAFD enforced thermodynamic rules, the gap between the external optimizer and NMF resulted in a solution that is far from the ground truth (see Fig.7). In contrast, DRNet almost exactly recovered the ground truth solution by seamlessly integrating pattern recognition, reasoning, and prior knowledge, including the novelty of explicitly incorporating the stick pattern information. DRNets solved the Bi-Cu-V oxide system, producing valid crystal structures and significantly outperforming IAFD w.r.t. reconstruction error. In addition, none of the IAFD phases matched the ICDD stick patterns, indicating that the Bi-Cu-V oxide system is beyond IAFD’s capabilities. Materials science experts thoroughly checked DRNet’s Bi-Cu-V-O solution, and approved it. They were particularly excited about the results given that the phase map for the **Bi-Cu-V-O system was previously unknown**, despite their considerable efforts.

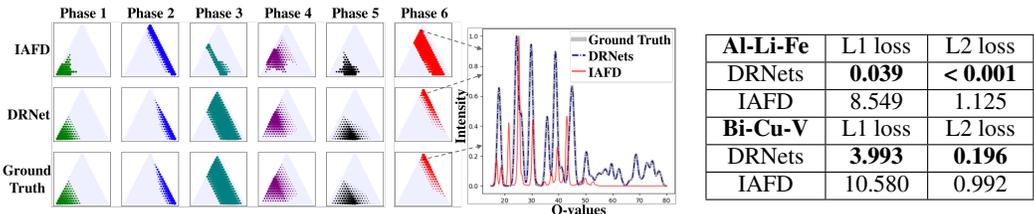


Figure 7: Comparison of phase concentration in Al-Li-Fe oxide system estimated by IAFD and DRNets. Each dot represents an XRD measurement whose size is proportional to the estimated phase concentration. DRNet’s phase patterns closely match the ground truth in contrast to IAFD’s (see e.g., phase 6, right panel).

Table 2: DRNets outperform IAFD both on L1 and L2. Num. of XRDs: Al-Li-Fe 231; Bi-Cu-V 353; Num. of stick patterns: Al-Li-Fe 159; Bi-Cu-V 100.

Combinatorial Problems: As a proof of concept of how DRNets can encode standard combinatorial problems, we solve 9-by-9 Sudoku puzzles and Boolean satisfiability problems (SAT), using a 3-layerfully-connected network as our encoder and the reasoning modules. We generated 10,000 9-by-9 Sudoku puzzles with 24 to 32 clues [Gordon Royle, 2014] (e.g., see Fig.1) and 10,000 satisfiable random 3-SAT instances with the hardest ratio (#clauses/#literals = 4.3) [Mitchell et al., 1992]. We compared DRNets with the supervised deep learning state of the art: Recurrent Relational Networks (RRNets) Palm et al. [2017], NeuroSAT Selsam et al. [2018] (SAT) and PDP [Amizadeh et al., 2019] (SAT). DRNets, *without supervision*, outperformed all supervised deep learning models (see Table 3).

Instances (10,000)	DRNets	DRNets + Restart	NeuralSAT	PDP	RRNets
3-SAT n=30 m=129	81.0% (4min)	99.0% (33min)	45.5% (2min+1hr)	78.9% (5min+2hr)	NA
3-SAT n=50 m=215	63.3% (7min)	94.0% (47min)	26.1% (3min+1hr)	62.2% (8min+2hr)	NA
3-SAT n=100 m=430	34.7% (17min)	77.9% (2hr)	4.7% (5min+1hr)	31.4% (2hr+2hr)	NA
9x9 Sudoku	99.5% (1hr)	99.8% (1hr)	NA	NA	99.6% (1min + 1day)

Table 3: Percentage of instances solved for 3-SAT (hardest ratio $m/n = 4.3$) and standard 9x9 Sudoku (24 to 32 known cells). We show the "test time + training time" for supervised baselines and the "solving time" for our unsupervised DRNets. n, m denote the number of literals and clauses. NA, not applicable. DRNets, without supervision, outperform the supervised state of the art.

Finally, we stress that our main goal is to tackle problems that combine deep learning and reasoning, such as de-mixing Sudokus or crystal structure phase mapping, as opposed to competing with pure, highly specialized state-of-the-art SAT solvers that can solve larger 3-SAT instances than the ones reported here. Nevertheless, our results show that DRNets can encode a broad range of combinatorial constraints and prior knowledge and effectively combine deep learning with reasoning.

See supplementary materials for further details on DRNets’ model and experimental results.

5 Conclusions and future work

We propose DRNets, a powerful end-to-end framework that combines deep learning with logical and constraint reasoning for solving complex tasks. DRNets outperform the state of the art for demixing MNIST Sudokus and crystal-structure phase mapping, solving previously unsolved systems substantially beyond the reach of other methods and materials science experts' capabilities. DRNets also outperform the deep-learning state of the art for solving standard Sudokus and 3-SAT. While we illustrate the potential of DRNets with unsupervised settings, it is straightforward to impose supervision into DRNets. Future research includes exploring DRNets for incorporating other types of constraints, prior knowledge, and objective functions, for other applications.

Acknowledgments

This work was supported by NSF awards CCF-1522054 and CNS-0832782 (Expeditions), CNS-1059284 (Infrastructure), and IIS-1344201 (INSPIRE); ARO award W911-NF-14-1-0498 and W911NF-17-1-0187 for the computational experiments (DURIP); AFOSR Multidisciplinary University Research Initiatives (MURI) Program FA9550-18-1-0136; an award from Toyota Research Institute; and US DOE Award No. de-sc0004993.

References

- Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.
- Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. In *Advances in Neural Information Processing Systems*, pages 5360–5370, 2017.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.
- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- Stefano Ermon, Ronan Le Bras, Santosh K Suram, John M Gregoire, Carla P Gomes, Bart Selman, and Robert B Van Dover. Pattern decomposition with complex combinatorial constraints: Application to materials discovery. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of sat problems. In *AAAI*, volume 92, pages 459–465, 1992.
- Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *Advances in neural information processing systems*, pages 25–32, 2004.
- Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049, 2010.
- Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*, 2016a.
- Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.
- Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679, 2016b.
- Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Weakly supervised transfer for 3d human pose estimation in the wild. In *IEEE International Conference on Computer Vision*, volume 206, page 3, 2017.
- Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, pages 186–201. Springer, 2016.
- Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.
- Di Chen, Yexiang Xue, and Carla P Gomes. End-to-end learning for the deep multivariate probit model. *arXiv preprint arXiv:1803.08591*, 2018.
- Jay Yoon Lee, Michael Wick, Jean-Baptiste Tristan, and Jaime Carbonell. Gradient-based inference for networks with output constraints. *arXiv preprint arXiv:1707.08608*, 2017.
- Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.
- Saïd Ladjal, Alasdair Newson, and Chi-Hieu Pham. A pca-like autoencoder, 2019.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

- Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.
- Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2011.
- Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017a.
- Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017b.
- Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, Xiaodan Liang, Lianhui Qin, Haoye Dong, and Eric Xing. Deep generative models with learnable knowledge constraints. *arXiv preprint arXiv:1806.09764*, 2018.
- Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d’horizon. *arXiv preprint arXiv:1811.06128*, 2018.
- Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.
- Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. In *Advances in Neural Information Processing Systems*, pages 539–548, 2018.
- Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L Dill. Learning a sat solver from single-bit supervision. *arXiv preprint arXiv:1802.03685*, 2018.
- Saeed Amizadeh, Sergiy Matushevych, and Markus Weimer. Pdp: A general neural framework for learning constraint satisfaction solvers. *arXiv preprint arXiv:1903.01969*, 2019.
- Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. *arXiv preprint arXiv:1809.05504*, 2018.
- Radhika Shivhare and Ch Aswani Kumar. On the cognitive process of abstraction. *Procedia Computer Science*, 89:243–252, 2016.
- Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- Carla P Gomes, Bart Selman, Henry Kautz, et al. Boosting combinatorial search through randomization. *AAAI/IAAI*, 98:431–437, 1998.
- Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Ronan Le Bras, Richard Bernstein, John M Gregoire, Santosh K Suram, Carla P Gomes, Bart Selman, and R Bruce Van Dover. Challenges in materials discovery—synthetic generator and real datasets. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- Junwen Bai, Johan Bjorck, Yexiang Xue, Santosh K Suram, John Gregoire, and Carla Gomes. Relaxation methods for constrained matrix factorization problems: solving the phase mapping problem in materials discovery. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 104–112. Springer, 2017.
- Gordon Royle. Minimum sudoku, 2014. URL <http://staffhome.ecm.uwa.edu.au/~00013890/sudokumin.php>.
- Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks for complex relational reasoning. *arXiv preprint arXiv:1711.08028*, 2017.