Deep Conditional Generative Models for Heuristic Search on Graphs with Expensive-to-Evaluate Edges

Brian Hou and Siddhartha Srinivasa

***Abstract*— Collision checking in motion planning is expensive, requiring lazy motion planning algorithms to explicitly reason about when edges should be evaluated. These algorithms defer collision-checking an edge until that edge is potentially along the optimal path. However, they often do not consider correlations between edge collision statuses. We present Edge-Collision Conditional Variational Auto-encoder (EC-CVAE), a model for learning these correlations purely from data without guidance from additional features (e.g. distance in task or configuration space). As more edges are evaluated, EC-CVAE more accurately predicts the collision statuses of the remaining edges. We show empirically that combining EC-CVAE with a lazy motion planning algorithm can reduce the number of edge evaluations needed to compute the optimal path.**

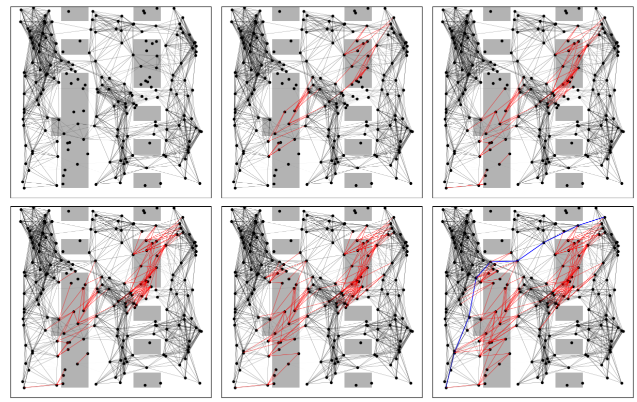

Fig. 1: Progress of LazySP using the EC-CVAE selector (Section V-C). Using EC-CVAE focuses expensive edge evaluations on edges that are likely to be in collision (red), quickly allowing the lazy motion planning algorithm to find the optimal path.

I. Introduction

To mitigate the bottleneck of collision checking, lazy motion planning algorithms defer collision evaluations between two configurations until that edge is potentially along the optimal path [3], [6], [7]. These planning algorithms maintain two sets of edges: *evaluated* edges (either VALID or INVALID) and *unevaluated* edges (optimistically assumed to be VALID). When deciding between several unevaluated candidate edges, edge-evaluation heuristics can focus collision checks on edges that have high probability of collision and quickly invalidate them.

Our key insight is that edge labels are highly correlated, enabling few edge evaluations to provide information about many edges. In addition, obstacles in real-world environments are rarely randomly placed: they often have some recurring structure that can be learned and exploited.

We model the validity of each edge as a Bernoulli random variable. Naïvely estimating the conditional distributions for all possible subsets of evaluated edges requires a table that is exponential in the number of edges in the graph. Instead, we train a conditional variational auto-encoder (CVAE) [13] to efficiently predict these edge collision probabilities, conditioned on the evaluated edge labels. This CVAE approach combines two sources of data to inform these probability estimates: *online observations* of edges that have been evaluated in the current planning environment and *offline examples* of similar planning environments where all edges in the graph have been evaluated. Using the CVAE predictions as an edge-evaluation heuristic enables a lazy motion planning algorithm to quickly find the optimal path.

Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA 98195

II. Related Work

Deep generative models such as variational auto-encoders (VAEs) [8] and generative adversarial networks (GANs) [5] use the expressiveness of deep models to model complex distributions of data. Deep conditional generative models (e.g. conditional VAEs [13]) enable further control over the sampled outputs by sampling from a different conditional distribution depending on the values of the conditioned variables. These approaches have been applied in tasks such as synthesis of realistic images and speech.

Our problem of predicting edge collision probabilities based on evaluated edge labels is similar to the problem of image completion, where masked pixels of an image must be recovered based on the remaining visible pixels. However, many of these algorithms (e.g. [12], [13]) decide on a fixed mask size to recover before training the network, making them unsuitable for our problem setting where the number of edges to predict varies. Other approaches based on modified convolutions are capable of recovering irregular masks [9], [14]. However, this cannot be directly applied to our problem; since vertices and edges can be arbitrarily ordered, there may be little correlation between a missing entry in the adjacency matrix and its neighbors.

Several machine learning approaches have been proposed to predict the validity of unevaluated edges. Esposito and Wright [4] argue that the adjacency matrix for a roadmap can be decomposed into the sum of a low-rank matrix and a sparse error matrix, and propose a convex optimization procedure to recover the unknown entries. This method attempts to recover the edge validities for all edges between any pair of vertices, while we assume that we only need to recover a fixed subset of edges. Their optimization-based approach cannot be used to sample different adjacency matrices that are consistent with the evaluated edges. Pan et al. [11] use point-to-point and line-to-point approximate nearest-neighbor queries to efficiently perform probabilistic collision checking for configurations and edges based on previously evaluated configurations. This approach only incorporates the validity of nearby points in configuration space to predict edge validities; however, points that are far apart in configuration space may still be informative, e.g. if they are close in task space. Furthermore, neither of these approaches can incorporate offline training examples to improve predictions.

Planning with expensive collision-checking is a well-studied problem in motion planning. We focus on the paradigm of lazy motion planning, where an edge is only evaluated if it is potentially along the optimal path. Lazy-PRM* and Lazy-RRG* are asymptotically-optimal anytime lazy motion planning algorithms which only evaluate an edge if it would result in a shorter path [7]. The LazySP class of algorithms continuously re-solve the shortest path problem assuming that unevaluated edges are collision-free while lazily eliminating edges as they are evaluated to be in collision, replanning until the optimal collision-free path is found [3]. Lazy Receding Horizon A* balances the LazySP tradeoff between planning time and collision-checking time [10]. These algorithms also do not incorporate offline training examples to decide which edges to evaluate.

Several motion planning algorithms assume access to edge collision probabilities. Haghtalab et al. bound the number of edge evaluations of the LazySP class of algorithms in this probabilistic setting and prove that LazySP is asymptotically optimal [6]. POMP is an anytime motion planning algorithm that leverages edge collision probabilities to quickly find a collision-free path with few edge evaluations, then balances path length and path collision probability to find shorter paths [1]. The BISECT algorithm uses the framework of Decision Region Determination to find a feasible path from a library of paths, computing priors on edge collision probabilities from offline training examples [2]. Our CVAE approach provides an alternative for generating these probabilities, so it is complementary to these algorithms.

III. PROBLEM STATEMENT

In the shortest path problem, we assume that we are given a graph $G = (V, E)$ in configuration space and desired start and goal vertices $v_s, v_g \in V$. Traversing each edge e in the graph incurs a cost of $w(e)$. The optimal path from the start to goal $p^*(v_s, v_g)$ minimizes the total cost of edges in the path. While the set of edges E is known and fixed, we are only given a lower bound for each edge weight $\hat{w}(e)$ (e.g. Euclidean distance). The exact weight $w(e)$ is unknown until the edge is evaluated for collisions: a collision-free edge has cost $\hat{w}(e)$ and an edge in collision has infinite cost. Since each edge evaluation is expensive, we wish to find the optimal path while minimizing the number of edge evaluations.

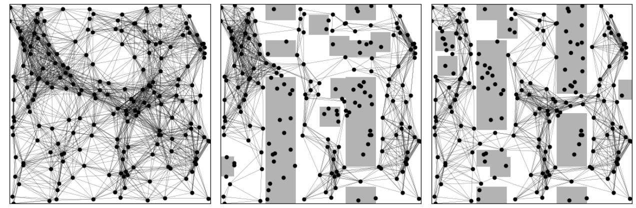

Fig. 2: Example environments from the TwoWall training dataset. The left figure is the graph that is shared by all environments in the dataset. Obstacles (gray) are for visualization purposes only; their positions are not provided as features to EC-CVAE.

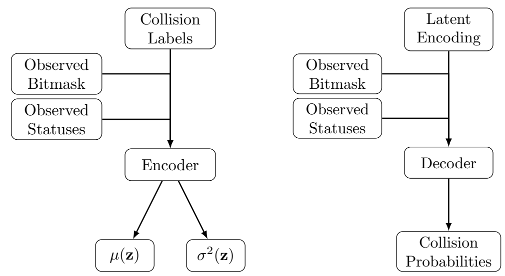

Fig. 3: Edge-Collision Conditional Variational Auto-encoder (EC-CVAE) encoder and decoder models. The collision labels, observed bitmask, observed statuses, and collision probabilities are all $|E|$-dimensional vectors, where $|E|$ is the number of edges in the graph.

We build on the LazySP framework proposed by Dellin and Srinivasa [3]. LazySP maintains an optimistic graph where an edge is assumed to be collision-free (i.e. edge weight is exactly $\hat{w}(e)$) until it has been evaluated to be in collision. It computes the shortest path on the optimistic graph, then selects an edge along this candidate path to evaluate according to an *edge-selector* function. If an edge along the candidate path is evaluated to be in collision, a new candidate is proposed based on the updated optimistic graph. If all edges along a candidate path are evaluated to be collision-free, it is the optimal path $p^*(v_s, v_g)$.

The choice of edge selector can have a drastic effect on the number of edges that LazySP evaluates. Our goal is to develop a selector that reduces edge evaluations by predicting which edges are likely to be in collision, based on previously evaluated edges.

While developing our data-informed selector, we assume that there is a single explicit graph with a fixed list of edges, which will be used for all environments. We assume that there are example environments where all edges in the graph have been collision-checked, which will be used as training data for our selector. Different subsets of edges will be in collision depending on the configuration of the obstacles in each environment.

IV. EDGE-COLLISION CONDITIONAL VARIATIONAL AUTO-ENCODER

Variational auto-encoders (VAEs) [8] encode high-dimensional data $\mathbf{x}$ to parameters for a low-dimensional latent random variable $\mathbf{z}$, where the latent distribution is chosen to be easy to sample from (e.g. $\mathcal{N}(0, I)$). Samples from the latent distribution are then decoded to reconstruct the original data. To sample from the resulting generative model, samples can be directly drawn from the chosen latent distribution and decoded. The VAE objective function can be viewed as minimizing the Kullback-Leibler divergence between the variational and true posterior distribution. Computing this divergence is intractable, but minimizing the divergence is equivalent to maximizing the tractable Evidence Lower Bound (ELBO). Conditional variational auto-encoders [13] provide an additional variable $\mathbf{o}$ to both the encoder and decoder networks, enabling the networks to learn different conditional distributions for each $\mathbf{o}$.

In this work, we describe our Edge-Collision Conditional Variational Auto-encoder model (EC-CVAE) for predicting edge collision probabilities based on learned correlations with previously evaluated edges. The data to reconstruct is the binary vector of ground-truth edge collision statuses $\mathbf{x} \in \mathbb{R}^{|E|}$ where 1 is in collision and 0 is collision-free. The observation $\mathbf{o}$ consists of the statuses of edges that have been evaluated. This is represented as two vectors in $\mathbb{R}^{|E|}$: a binary vector that indicates whether each edge has been evaluated, and a vector of statuses (1 for collision, 0 for collision-free, and 0.5 for unevaluated).

We believe this model will scale effectively to graphs in higher-dimensional configuration spaces since the network size is defined by the number of edges. In Section V-B, we show that EC-CVAE (Fig. 3) naturally learns correlations between edge collision statuses without any additional features (e.g. task or configuration space distance).

V. Experiments

We evaluate EC-CVAE by considering a variety of 2D geometric motion-planning problems [2]. We focus on these simply for ease of visualization; our model depends on the number of edges in the graph, not the dimension of the configuration space.

A. Experimental Setup

Both the encoder and decoder network in EC-CVAE have two fully-connected layers of 256 hidden units. We selected a latent dimensionality of 10 by comparing the ELBO of different latent dimensionalities on the validation set, averaged over 3 random seeds. Each EC-CVAE model was trained for 100 epochs.[1] In one epoch, we sample an edge observation probability (uniformly between 0 and 1) for each environment in the training dataset. Each edge is observed independently according to that probability to generate the observation $\mathbf{o}$ that is passed into the network.

B. Reconstructions and Samples

In Table I, we compare the reconstruction performance of EC-CVAE on five 2D geometric motion-planning datasets from Choudhury et al. [2] with 1% and 5% of edges evaluated. However, since there may be many possible samples

Dataset	% Evaluated	Norm. Loss	% Incorrect
OneWall	1%	0.256 ± 0.006	12.3%
$\|E\| = 923$	5%	0.228 ± 0.006	10.5%
TwoWall	1%	0.111 ± 0.005	4.2%
$\|E\| = 2524$	5%	0.097 ± 0.005	3.7%
Forest	1%	0.391 ± 0.005	18.8%
$\|E\| = 2524$	5%	0.345 ± 0.004	15.9%
SolidOneWall	1%	0.059 ± 0.002	2.3%
$\|E\| = 1689$	5%	0.056 ± 0.002	2.1%
SolidTwoWall	1%	0.051 ± 0.002	2.3%
$\|E\| = 1689$	5%	0.046 ± 0.002	2.0%

TABLE I: Reconstruction performance of environments in the test set when 1% and 5% of edges have been evaluated. The reconstruction error is the average binary cross-entropy loss between the predicted edge collision probabilities and ground-truth edge collision statuses. Losses are normalized by the number of edges in the graph. Percent incorrect when edge collision probabilities are rounded to 0 or 1 and compared to the ground truth labels.

Selector	Offline Examples	Online Observations
Forward		
Prior	✓	
EC-CVAE	✓	✓

TABLE II: Comparison of data considered by each selector. Prior uses offline examples to empirically estimate collision probabilities. EC-CVAE trains the model on offline examples and generates samples based on online observations.

that are consistent with the observed edge labels (e.g. when no edges have been evaluated), it may be impossible to recover the exact edge collision probabilities. As a result, this is only a proxy for EC-CVAE performance.

We visualize EC-CVAE samples for the TwoWall dataset with 1% of edges evaluated in Fig. 4. The samples are not perfect: edges that have already been evaluated to be in collision can be predicted to be collision-free and vice versa. However, they do capture some correlations between edges. For example, middle edges through the right wall are usually predicted to be in collision or collision-free as a group. In the top row of samples, since some of those edges have already been evaluated to be in collision, the group is usually predicted to be in collision.

C. Planning Results

To evaluate whether EC-CVAE reduces the number of edges evaluated by LazySP, we compare the performance of three selectors on the TwoWall dataset (Fig. 5). The Forward selector returns the unevaluated edge closest to the beginning of the path. The Prior and EC-CVAE selectors return the edge with the highest probability of collision; Prior chooses based on the prior probability that each edge is in collision, while EC-CVAE updates its probability predictions as edges are evaluated.[2]

[1] Training on a 2017 MacBook Pro (2.3 GHz Intel Core i5 processor, 8 GB of memory) takes 3 minutes for the OneWall environment and 6 minutes for the TwoWall environment.

[2] Prior is similar to the MaxTally selector proposed by Choudhury et al. [2].

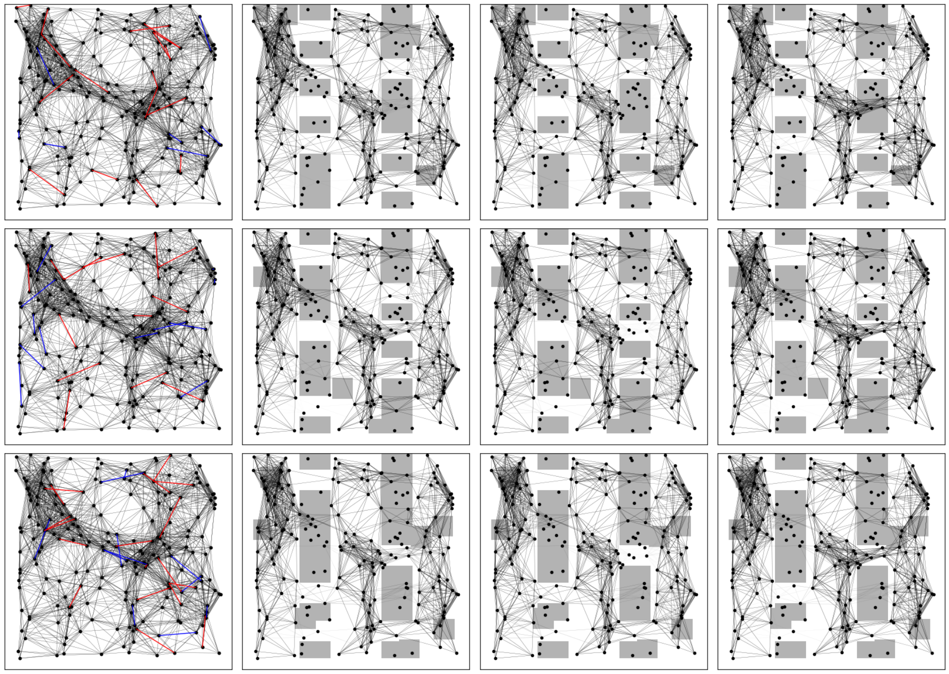

Fig. 4: EC-CVAE samples for the TwoWall dataset when 1% of edges have been evaluated. Column 1 visualizes the collision statuses of evaluated edges (VALID in blue, INVALID in red). Columns 2-4 visualize EC-CVAE samples, where darker edge lines mean that EC-CVAE is more confident that the edge is collision-free. EC-CVAE learns some interesting correlations between edges: middle edges through the right wall are usually predicted to be in collision or collision-free as a group. In the top row of samples, since some of those edges have already been evaluated to be in collision, the group is usually predicted to be in collision.

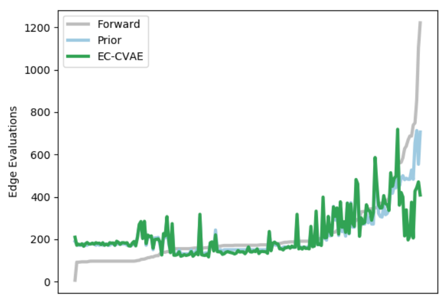

Fig. 5: Edge evaluation results for LazySP on different environments from the TwoWall dataset. Results are ordered by increasing number of edge evaluations performed by the FORWARD selector as a proxy for problem difficulty.

On most environments in this dataset, the PRIOR and EC-CVAE selectors perform comparably. However, on the environments where FORWARD evaluates many edges, EC-CVAE shows improved performance over PRIOR. Fig. 1 shows snapshots of the progress that the EC-CVAE selector makes on one of those environments.

VI. Discussion and Future Work

We have presented an approach for learning a conditional generative model for edge collision probabilities and leveraging that model to inform an edge-evaluation heuristic for a lazy motion planning algorithm. The LazySP framework is beneficial for focusing evaluation on current candidate paths, but other motion planning algorithms that leverage black-box edge collision probabilities will also find EC-CVAE useful.

In this work, we have focused on the setting of a single graph that is used for many environments. Our current world representation as a vector of edge collision statuses makes it challenging for our learned model to generalize from the original graph to different graphs in similar environments. However, we believe that a similar generative approach with a different world representation could make it easier to leverage different graphs.

One major drawback with EC-CVAE and other CVAE-based generative models is that evaluated edges are not hard constraints on the output of the network: edges that have already been evaluated to be valid may still be predicted to be invalid and vice versa. One possible method for addressing this is by having an asymmetric cost function that penalizes reconstruction error on observed edges more strongly than unobserved edges. In addition, sampling edge collision probabilities from an EC-CVAE often results in very similar samples.

In spite of these issues, however, CVAEs work surprisingly well. We believe that the existing EC-CVAE model can be combined with a more sophisticated edge selector strategy in the LazySP framework to further reduce the number of edges evaluated. For example, it is possible to efficiently implement a selector that maximizes myopic value of information instead of edge collision probability. Furthermore, we believe that EC-CVAE can scale effectively to higher-dimensional configuration spaces since the network size is defined by the number of edges in the graph. We will validate these hypotheses and consider alternative generative models in future work.

Acknowledgements

This work was supported by a NASA Space Technology Research Fellowship (#80NSSC17K0137), the National Institute of Health R01 (#R01EB019335), National Science Foundation CPS (#1544797), National Science Foundation NRI (#1637748), the Office of Naval Research, the RCTA, Amazon, and Honda.

References

[1] S. Choudhury, C. Dellin, and S.S. Srinivasa. Pareto-optimal search over configuration space beliefs for anytime motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[2] S. Choudhury, S. Javdani, S.S. Srinivasa, and S. Scherer. Near-optimal edge evaluation in explicit generalized binomial graphs. In *Advances in Neural Information Processing Systems*, 2017.

[3] C. Dellin and S.S. Srinivasa. A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors. In *International Conference on Automated Planning and Scheduling*, 2016.

[4] J.M. Esposito and J.N. Wright. Matrix completion as a post-processing technique for probabilistic roadmaps. In *Workshop on the Algorithmic Foundations of Robotics*, 2016.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[6] N. Haghtalab, S. Mackenzie, A.D. Procaccia, O Salzman, and S.S. Srinivasa. The Provable Virtue of Laziness in Motion Planning. In *International Conference on Automated Planning and Scheduling*, 2018.

[7] K. Hauser. Lazy collision checking in asymptotically-optimal motion planning. In *IEEE International Conference on Robotics and Automation*, 2015.

[8] D.P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[9] G. Liu, F.A. Reda, K.J. Shih, T.C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018.

[10] A. Mandalika, O. Salzman, and S.S. Srinivasa. Lazy Receding Horizon A* for Efficient Path Planning in Graphs with Expensive-to-Evaluate Edges. In *International Conference on Automated Planning and Scheduling*, 2018.

[11] J. Pan, S. Chitta, and D. Manocha. Faster sample-based motion planning using instance-based learning. In *Workshop on the Algorithmic Foundations of Robotics*, 2012.

[12] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A.A. Efros. Context encoders: Feature learning by inpainting. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[13] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 2015.

[14] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T.S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.