CausalAF: Causal Autoregressive Flow for Safety-Critical Scenes Generation

Anonymous Authors[1]

Abstract

Goal-directed generation, aiming for solving downstream tasks by generating diverse data, has a potentially wide range of applications in the real world. Previous works tend to formulate goal-directed generation as a purely data-driven problem, which directly approximates the distribution of samples satisfying the goal. However, the generation ability of preexisting work is heavily restricted by inefficient sampling, especially for sparse goals that rarely show up in off-the-shelf datasets. For instance, generating safety-critical traffic scenes with the goal of increasing the risk of collision is critical to evaluate autonomous vehicles, but the rareness of such scenes is the biggest resistance. In this paper, we integrate causality as a prior into the safety-critical scene generation process and propose a flow-based generative framework – *Causal Autoregressive Flow (CausalAF)*. CausalAF encourages the generative model to uncover and follow the causal relationship among generated objects via novel causal masking operations instead of searching the sample only from observational data. Extensive experiments on three heterogeneous traffic scenes illustrate that *CausalAF* requires much fewer optimization resources to effectively generate goal-directed scenes for safety evaluation tasks.

1. Introduction

Deep generative models (DGMs) have shown their powers for data generation in several domains. Recently, people have been weary of random generation and turned to generating goal-directed samples useful for downstream tasks. Standing on the top of successful DGMs, goal-directed generation demonstrates potentiality in molecule (Shi et al., 2020) and natural language (Mollaysa et al., 2020) areas, which is usually formulated as shifting the generative distribution to satisfy specific goals.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

One typical application of goal-directed generation is generating traffic scenes, which is a universally acknowledged way to evaluate autonomous vehicles (Riedmaier et al., 2020). Rare but significant, safety-critical scenes are extraordinarily important for the evaluation. Taking the *safety-critical* scene as a goal, such a generation task is challenging since we need to simultaneously consider scene realism to avoid conjectural scenes that will never happen in the real world, as well as the safety-critical level which are indeed rare compared with ordinary scenes. In addition, generating reasonable threats to vehicles' safety can be inefficient if the model purely relies on the correlation of observation, as the safety-critical scenes are rare and follow certain fundamental physical principles.

Existing work (Engel et al., 2017) searches in the latent space of generative model to build scenes that satisfy downstream requirements. The biggest challenge is that ordinary scenes may dominate the latent space while safety-critical samples are ignored as "outliers". Another approach (Tripp et al., 2020) is to retrain the model during the searching to avoid forgetting the high-quality but rare data. However, the efficiency could still be unacceptably low due to the sparsity of qualified samples. In contrast, humans are good at abstracting the causation beneath the observations with prior knowledge, which lights up a new direction towards causal generative models.

In this paper, we build a goal-directed generative model with causal priors that are accessible in many applications. We model the causality as a directed acyclic graph (DAG) named causal graph (CG) (Pearl, 2009). To facilitate CG in the downstream tasks, we propose the Behavioral Graph (BG), which can be regarded as instances of CG (Grünbaum, 1952), for interactive and dynamic scenes representation. The graphical representation of both graphs makes it possible to use the BG to unearth the causality given by CG. We propose the first generative model that integrates causation into the graph generation task and name it *CausalAF*. To connect BG and CG at the graph level, we propose two types of causal masks – Causal Order Masks (COM) and Causal Visibility masks (CVM). COM modifies the node order for node generation, and CVM removes irrelevant information for edge generation.

For a better explanation, we consider a running example of a traffic scene. When the vision of the autonomous vehicle a is clear, a can easily see the pedestrian c crossing the road then decelerate in advance. However, if another vehicle b is parked in the middle between a and c, the vision of a will be blocked, making a have less time to brake and more likely to collide c. This example may take autonomous driving vehicles millions of hours to collect (Feng et al., 2021), which is challenging for real-world applications. However, when we use a generative model to create such a scene, it will not consider the causality but try only to memorize the location of all objects then generate adversarial examples (Goodfellow et al., 2014b). Consequently, the generated scene may not cause any risk if the objects are slightly different.

Overall, we show the diagram of goal-directed generation with *CausalAF* in Fig. 1 and we summarize our contributions below:

- We proposed a causal generative model named *CausalAF* that integrates causal graphs and temporal graphs for safety-critical scene generation.
- We designed two novel mask operators to reliably integrate causation order and causation visibility into the flow-based generation procedure.
- We showed *CausalAF* demonstrates dramatic improvement in efficiency and generalizability on three standard traffic settings compared with purely data-driven goal-directed baseline.

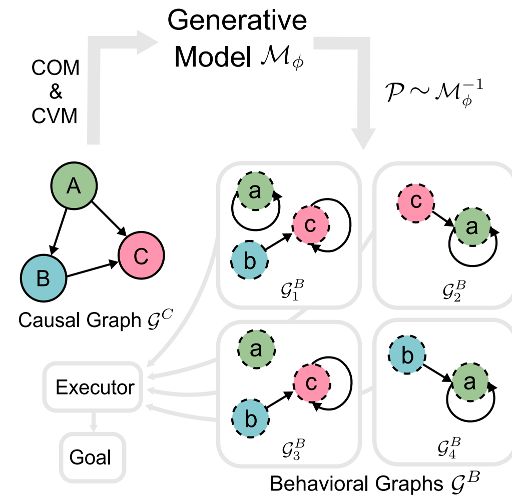

Figure 1. Diagram of proposed *CausalAF* framework.

2. Related Work

2.1. Goal-directed Generative Models

DGMs, such as Generative Adversarial Networks (Goodfellow et al., 2014a) and Variational Auto-encoder (Kingma & Welling, 2013), have shown powerful capability in randomly data generation tasks (Brock et al., 2018). Thanks to the boom of diverse DGMs, goal-directed generation methods are widely used in many applications (Mollaysa et al., 2020). One line of research leverages conditional GAN (Mirza & Osindero, 2014) and conditional VAE (Sohn et al., 2015), which take as input the conditions or labels during the training stage. Another line of research injects the goal into the model after the training. (Engel et al., 2017) proposes a latent space optimization framework that finds the samples by searching in the latent space. This spirit is also adopted in other fields: (Mollaysa et al., 2019) finds the molecules that satisfy specific chemical properties, (Abdal et al., 2020) searches in the latent space of StyleGAN (Karras et al., 2019) to obtain targeted images.

Recent works combine the advantages of the above two lines by retraining the generative model during the search. To expand the area of the desired region in the latent space, (Tripp et al., 2020) iteratively updates the high-quality samples and retrains the model weights. (Shi et al., 2020) pre-trains the generative model and optimize the sample distribution with reinforcement learning algorithms. This paper enhances the generalizability and efficiency by leveraging causation graphs so that it is applicable to rare safety-critical scenes.

2.2. Safety-critical Traffic Scene Generation

Traditional traffic scene generation algorithms sample from pre-defined rules and grammars, such as probabilistic scene graphs (Prakash et al., 2019) and heuristic rules (Dosovitskiy et al., 2017). In contrast, DGMs (Devaranjan et al., 2020; Tan et al., 2021; Ding et al., 2018; 2020) are recently used to learn the distribution of objects to construct diverse scenes. There are two lines of work. One is to directly search for the adversarial scenes. (Zeng et al., 2019) modifies the light condition. (Alcorn et al., 2019; Xiao et al., 2019; Jain et al., 2019) manipulate the pose of objects in traffic scenes. (Tu et al., 2020; Abdelfattah et al., 2021) adds objects on the top of existing vehicles to make them disappear, (Sun et al., 2020) creates a ghost vehicle by adding an ignorable number of points, and (Ding et al., 2021b) generates the layout of the traffic scene with a tree structure integrated with human knowledge. Another line of research generates the risky scenes while also considering the likelihood of occurring of the scenes in the real world, which requires a probabilistic model of the environment. (Zhao et al., 2017; O'Kelly et al., 2018; Arief et al., 2021) used various importance sampling approaches to generate risky but probable scenes. (Ding et al., 2020) merges the naturalistic and collision datasets with conditional VAE to generate near-misses. (Ding et al., 2021a) uses reinforcement learning to search for risky cyclist encounters for victim cars with a penalty of rarity. Compared with purely probabilistic methods, *CausalAF* method may have better generalization, data efficiency, and statistically robust against sparse data as it not only learns Bayesian models but also capture the causation of collisions.

2.3. Causal Generative Models and Representation Learning

The research of causality, mainly described with probabilistic graphical models-based language (Pearl, 2009), is usually divided into two aspects: causal discovery tries to find the underlying mechanism from the observational and interventional data. In contrast, causal inference extrapolates the given causality to solve new problems. Discovering the causal graph has been prevalent for several decades. (Zhu et al., 2019) proposed a flexible and efficient RL-based method to search over the DAGs space for the best causal graph that fits the dataset. A toolbox named NOTEARs is proposed in (Zheng et al., 2018) to learn causal structure in a fully differentiable way, which drastically reduces the complexity caused by combinatorial optimization. (Heckerman et al., 1995) show the identifiability of learned causal structure from interventional data, which is obtained by manipulating the causal system under interventions.

Recently, causality has been introduced into DGMs to learn the cause and effect with representation learning. CausalGAN (Kocaoglu et al., 2017) captures the causation between labels by training the generator with the causal graph as a prior, which is very similar to our setting. In CausalVAE (Yang et al., 2021), the authors disentangle latent factors by learning a causal graph from data and corresponding labels. Previous work CAREFL (Khemakhem et al., 2021) also explored the combination of causation and autoregressive flow-based model and is used for causal discovery and prediction tasks.

3. Representation of Causation and Scenes

Our *CausalAF* is built upon the relation between the CG and the BG. We start by introducing the definition of these two types of graphs and the autoregressive generation process of the BG.

3.1. Causal Graph and Behavioral Graph

The causal graph is defined over m random variables $\{x_1, ..., x_m\}$. The variables in this vector forms a DAG $\mathcal{G}^C = (V^C, E^C)$. $V^C \in \{0,1\}^{m\times n}$ is the node matrix and $E^C \in \{0,1\}^{m\times m}$ is the adjacency matrix with m nodes in n types. Each node i is associated with a random variable x_i. Each edge (i,j) represents a causal relation from variable x_i to x_j. For a DAG, there exists a (not necessarily unique) causal order of the nodes, such that the cause variable precedes the effect variable: $p(x_1, ..., x_n) = \prod_{j=1}^{n} p_j(x_j |\ \mathbf{pa}(x_j))$, where $\mathbf{pa}(x_j)$ represents the parent nodes for variable x_j. In this work, we assume $\mathcal{G}^C$ is fully accessible with human knowledge and experience for certain tasks.

We then define the Behavioral Graph $\mathcal{G}^B$ to represent objects in a dynamic and interactive scene. According to **Definition 1**, $\mathcal{G}^B$ works as a high-level planner for objects and controls their behaviors in the physical scene with interpretable edge meanings. A self-loop edge (i,i) represents that one object takes one action irrelevant to other objects (e.g., a car goes straight or turns left with no impact on other road users), while other edges (i,j) means object i takes one action related to object j (e.g., a car i moves towards a pedestrian j). The edge attributes represent the properties of actions. For instance, the attribute $[x, y, v_x, v_y]$ of one edge represents the 2-d position and velocity for agent nodes.

Definition 1. ***(Behavioral Graph)*** *Suppose there are n types of nodes and a scene have m objects. Then the Behavioral Graph $\mathcal{G}^B = (V^B, E^B)$ contains a node matrix $V^B \in \mathbb{R}^{m\times n}$ representing the categories of objects and an edge matrix $E^B \in \mathbb{R}^{m\times m\times(h_1+h_2)}$ representing the sequential interaction between objects, where h_1 is the number of edge types and h_2 is the dimension of edge attributes.*

3.2. Behavioral Graph Generation with Autoregressive Flow

Considering the directed acyclic nature of $\mathcal{G}^C$, we incorporate autoregressive flow models (AF) (Huang et al., 2018), which is a type of DGMs that sequentially generate nodes based on their predecessors to generate $\mathcal{G}^B$. It uses an invertible and differentiable transformation f to convert the observations $\boldsymbol{x}$ to a latent variable $\boldsymbol{z}$ that follows a base distribution $p_0(\boldsymbol{z})$ (e.g., Normal distribution). According to the change of variables theorem, we can obtain $p_{\boldsymbol{x}}(\boldsymbol{x}) = p_0(f^{-1}(\boldsymbol{x}))\left|\det\frac{\partial f^{-1}(x)}{\partial \boldsymbol{x}}\right|$. To increase the representing capability, we repeatedly substitute the variable for the new variable z_i and eventually obtain a probability distribution of $\boldsymbol{x}$ whose log-likelihood can be written as:

$$\log p(\boldsymbol{x}) = p_0(\boldsymbol{z}_0) - \sum_{i=1}^{K} \log\left|\det\frac{df_i}{d\boldsymbol{z}_{i-1}}\right| \quad (1)$$

In AF models, the transformation f construct $\boldsymbol{x}$ in a sequential way, which is naturally consistent with the construction of $\mathcal{G}^C$. To implement the function invertible f, we build a model $\mathcal{M}_\phi$ parametrized by ϕ . The inverse of $\mathcal{M}_\phi$, denoted as $\mathcal{M}_\phi^{-1}$, can be used to sample new data from Gaussian noises: $\boldsymbol{x} = \boldsymbol{z}_K = f_K^{-1} \circ f_{K-1}^{-1} \circ \cdots \circ f_0^{-1} = \mathcal{M}_\phi^{-1}(\boldsymbol{z}_0)$, where $\circ$ means the composition of two functions and $\boldsymbol{z}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Let $V^B_{[i,:]}$ and $E^B_{[i,j,:]}$ represent the node x_i

and edge (i, j) of $\mathcal{G}^B$ sampled from Gaussian distribution

$$
\begin{aligned}
V^B_{[i,:]} &\sim \mathcal{N}\left(\mu^v_i, (\sigma^v_i)^2\right) = \mu^v_i + \sigma^v_i \odot \epsilon \\
E^B_{[i,j,:]} &\sim \mathcal{N}\left(\mu^e_{i,j}, (\sigma^e_{i,j})^2\right) = \mu^e_{i,j} + \sigma^e_{i,j} \odot \epsilon
\end{aligned} \tag{2}
$$

where $\odot$ denotes the element-wise product. ϵ follows a Normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $[:]$ represents all elements in one dimension. In (2), variables μ^v_i, σ^v_i, $\mu^e_{i,j}$, and $\sigma^e_{i,j}$ are obtained from $\mathcal{M}_\phi$:

$$
\begin{aligned}
\mu^v_i, \sigma^v_i &= \mathcal{M}_\phi(V^B_{[0:i-1]}, E^B_{[0:i-1,:]}) \\
\mu^e_{i,j}, \sigma^e_{i,j} &= \mathcal{M}_\phi(V^B_{[0:i]}, E^B_{[0:i,0:j-1]})
\end{aligned} \tag{3}
$$

where $[0 : i]$ represents the elements from index 0 to index i. According to (3), the generation of the current node depends on all previous nodes and edges. Then the edges between current node and previous nodes are generated. Eventually, E^B will be an upper-triangular matrix since only the latter generated nodes have edges pointed to formerly generated nodes. This process is illustrated in Fig. 2.

4. Causal Autoregressive Flow (CausalAF)

Transferring the prior knowledge from $\mathcal{G}^C$ to $\mathcal{G}^B$ can be implemented by increasing the similarity between them. However, this similarity is not easy to calculate because it includes the directions between nodes and the input information of nodes. To solve this problem, we propose the *CausalAF* model with two causal masks, i.e., Causal Order Masks (COM) and Causal Visible Masks (CVM), that make the generated $\mathcal{G}^B$ follow the causal information given in $\mathcal{G}^C$. Particularly, COM is designed for regulating the order of the node generation, and CVM dynamically masks out irrelevant information during the edge generation.

Causal Order Masks The order is vital during the generation of $\mathcal{G}^C$ since we must ensure the cause is generated before the effect. To achieve this, we maintain a priority queue $\mathbb{Q}$ to store the valid node types for the current step. $\mathbb{Q}$ is initialized with $\mathbb{Q} = \{x_i | \, \mathbf{pa}(x_i) = \emptyset\}$, which means all nodes that do not have parent nodes are valid at the beginning. Then, in each node generation step, we update S by removing the generated node x_i and adding the child nodes of x_i. Notice that one node could have multiple parents; thus, we consider one node valid only if all of its parents have been generated. To encourage the model to generate nodes that satisfy the causal order, we use $\mathbb{Q}$ to create a k-hot mask $M^o(\mathcal{G}^C) \in \mathbb{R}^n$, where the element is set to 1 if it is corresponding to a valid node. Then, the type of next node x_i will be obtain by $v_i = \arg\max(M^o(\mathcal{G}^C) \odot \text{softmax}(V^B[i, :]))$, where $V^B[i, :]$ is the original node matrix obtained from $\mathcal{M}_\phi$ for node x_i. Intuitively, this mask reduces the probability of the invalid node types to 0 to ensure the generated node follows the correct order.

Causal Visible Masks Ensuring a correct causal order is still insufficient to represent the causality, which will be discussed in the later experiments. Thus, we further propose another type of mask called CVM. COM serves as a precondition for CVM in that it guarantees the existence of one node's parents before this node is ready to be generated. Otherwise, one node may lose prior information without knowing its causes.

At the step of generating edges for node x_i, we maintain the current generated graph with $\mathcal{G}^B(t) = (V^B(t), E^B(t))$, where t is the index for current step. Then, CVM is implemented with $M^x(\mathcal{G}^C) \in \mathbb{R}^{m\times n}$ and $M^e(\mathcal{G}^C) \in \mathbb{R}^{m\times m\times(h_1+h_2)}$ that satisfy

$$
\begin{aligned}
&M^x(\mathcal{G}^C)[j, :] = 0 \\
&M^e(\mathcal{G}^C)[:, j, :] = \mathbf{0} \\
&M^e(\mathcal{G}^C)[j, :, :] = \mathbf{0},\ \forall\{j \mid x_j \notin \mathbf{pa}(x_i)\}
\end{aligned} \tag{4}
$$

With these two masks, we can update $\mathcal{G}^B(t)$ before using it for next step by

$$
\begin{aligned}
V^B(t) &\leftarrow V^B(t) \odot M^x(\mathcal{G}^C) \\
E^B(t) &\leftarrow E^B(t) \odot M^e(\mathcal{G}^C)
\end{aligned} \tag{5}
$$

We illustrate an example of CVM in (c) of Fig. 2. Assume we are generating edges for node c. We need to remove node b since node B does not have edges to node C. After applying $M^x(\mathcal{G}^C)$ and $M^e(\mathcal{G}^C)$, we move the features of node c to the previous position of b. This permuting operation is important since the autoregressive model is not permutation invariant.

Goal-directed Optimization We then discuss the training of *CausalAF*. The target of goal-directed generation is to create samples satisfying a given goal, which is formulated as an optimization over objective function $\min_\phi \mathbb{E}_{\mathcal{G}^B\sim M_\phi^{-1}}[\mathcal{L}_g(\mathcal{G}^B)]$. Usually, the objective $\mathcal{L}_g$ contains non-differentiable operators (e.g., complicated simulation and rendering), thus we have to utilize black-box optimization methods to solve the problem. We consider a policy gradient algorithm named REINFORCE (Williams, 1992), which estimates the gradient from samples by

$$
\begin{aligned}
&\nabla_\phi \mathcal{L}_g(\mathcal{G}^B) \\
=&\mathbb{E}_{\mathcal{G}^B\sim M_\phi^{-1}}[\nabla_\phi \log M_\phi(\mathcal{G}^B)\mathcal{L}_g(\mathcal{G}^B)] \\
=&\frac{1}{N}\sum_{i=1}^{N}(\nabla_\phi \log M_\phi(\mathcal{G}^B_i)\mathcal{L}_g(\mathcal{G}^B_i))
\end{aligned} \tag{6}
$$

where N is the number of samples used for each iteration. Overall, the entire training algorithm is summarized in **Algorithm** 1.

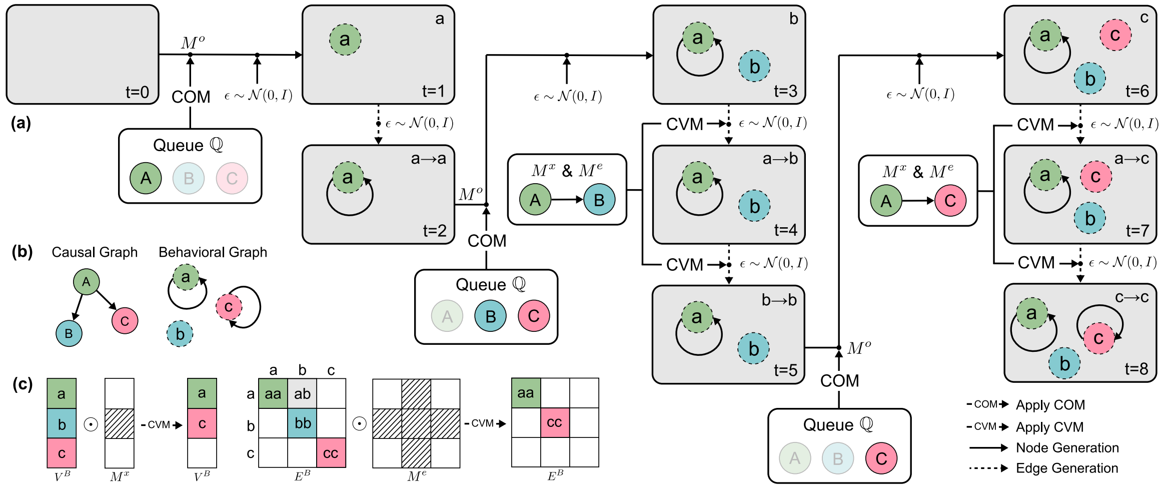

Figure 2. **(a)** The generation process of a Behavioral Graph. **(b)** The causal graph and Behavioral Graph used in the example of **(a)**. **(c)** The explanation of CVM when generating edges for c, where irrelevant node b is masked out in both V^B and E^B.

Algorithm 1 Training process of CausalAF

Input: Dataset $\mathcal{D}$, Causal Graph $\mathcal{G}^C$, Goal $\mathcal{L}_g$, Learning rate α, Maximum node number m
Output: The trained model $\mathcal{M}_\phi$
Initialize $\mathcal{M}_\phi$ by maximizing (1) on $\mathcal{D}$
while not converged **do**
  \\ Sample an BG from model $\mathcal{G}^B \sim M_\phi^{-1}$
  for $i < m$ **do**
    Sample a node $V^B[i, :]$ by (2)
    Calculate $M^o(\mathcal{G}^C)$ for COM
    Apply (4) to get the node type v_i
    Calculate $M^x(\mathcal{G}^C)$ and $M^e(\mathcal{G}^C)$ for CVM by (4)
    for $j < i$ **do**
      Apply CVM to nodes V^B and edges E^B by (5)
      Sample an edge $E^B[i, j, :]$ by (2)
    end for
  end for
  \\ Learn model parameters
  Calculate the likelihood $M_\phi(\mathcal{G}^B)$ of samples
  Execute $\mathcal{G}^B$ to get the goal objective $\mathcal{L}_g(\mathcal{G}^B)$
  Update parameters $\phi = \phi - \alpha \nabla_\phi \mathcal{L}_g(\mathcal{G}^B)$ via (6)
end while

5. Experiment

We evaluate *CausalAF* using three top pre-crash traffic scenes defined in (Najm et al., 2013) and (Van Ratingen et al., 2016). The benefit of the experimental setting is that humans usually have good intuitions of traffic scenes to examine the results. However, our empirical results show that it may not be trivial for the generative models to learn the underlying causality given the observational data, even if such causality seems understandable to humans. Particularly, we conduct a series of experiments to answer the question: whether there is a significant benefit to integrate causation into the generative models? We found that *CausalAF* outperforms the baseline and the advantages can be mainly attributed to the causation introduced by COM and CVM that eliminates irrelevant variables.

Simulator for typical Scenes We consider three safety-critical traffic scenes (shown in Fig. 3) that have clear causation. The $\mathcal{G}^C$ for each scenario is displayed on the upper right of the scene. These $\mathcal{G}^C$ are not necessarily unique for the scene, while they just hypothesize the potential causation.

- **Traffic-light**. One potential safety-critical event could happen when the traffic light T turns from green to yellow to give road right to an autonomous vehicle A. R runs the red light, colliding with with A perpendicularly. Here, A node is the parent for both T and R. T is also a parent for R because the risk vehicle follows the traffic light T.
- **Pedestrian**. A pedestrian P and an autonomous vehicle A are crossing the road in vertical directions. There also exists a static vehicle S parked by the side of the road. Then a potentially risky scene could happen when S blocks the vision of A and P. In this scene, A node is the parent for both P and S. S is also a parent for P since S determines the vision of P.
- **Lane-changing**. An autonomous vehicle A takes a lane-changing behavior due to a static car S parked in front of it. Meanwhile, a vehicle R drives in the opposite lane. When S blocks the vision of A, then A is likely to collide with R. In this scene, we make A node as the parent for both R and S. S is also a parent for R since the S determines the vision of P.

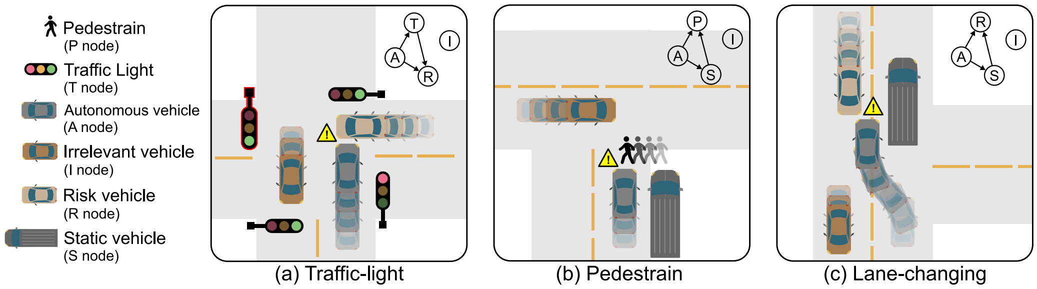

Figure 3. Three causal traffic scenes used in our experiments with corresponding causal graphs

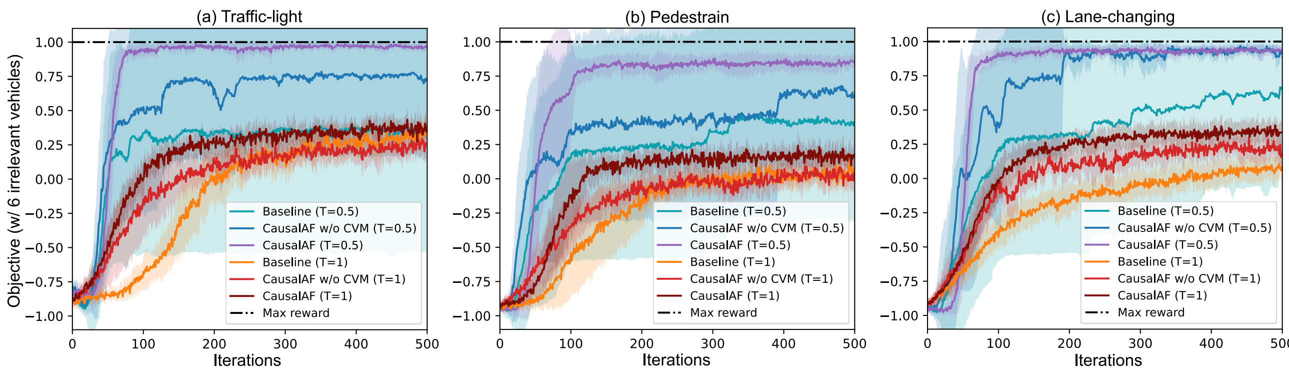

Figure 4. The training objective $\mathcal{L}_g(\mathcal{G}^B)$ of three scenes under two temperature settings.

We implement these scenes in a 2D simulator, where all agents have radar sensors and dynamic models. To avoid unrealistic collisions, the agent will brake if it detects any obstacles in front of it. In this case, the collision will not happen unless the radar of one agent is blocked and the distance is smaller than the braking distance. This setting is vital in that it avoids unrealistic collisions and makes the collision as sparse as in the real world. During the experiments, the goal-directed generative model firstly samples an $\mathcal{G}^B$. Then, the physical properties (e.g., position and velocity) defined in the generated $\mathcal{G}^B$ is executed in the simulator to create sequential scenes. After the execution, the simulator outputs the objective function $L_g(\mathcal{G}^B)$ as the simulation result.

Our goal is to generate risky scenarios that make collision happen for node A. Therefore, we set the object function to be a very sparse function: $\mathcal{L}_g(\mathcal{G}^B) = 1$ only if $\mathcal{G}^B$ causes collisions. Since generating goal-directed scenes is a new task, there are no existing methods to compare. We implement a baseline model with exactly the same structure as CausalAF without considering the causation during generation to represent data-driven generative models. We also compare with a model without CVM to conduct ablation studies.

Results and discussion We show the training objectives of three scenes in Fig 4. Notice that there are two temperatures $T = 0.5$ and $T = 1.0$ for all methods, which is use to control the sampling variance $\epsilon \sim \mathcal{N}(0, T)$. A large temperature provides strong exploration but also causes slow convergence. In all three scenes, *CausalAF* outperforms baseline, and the gap is more significant under $T = 1.0$ setting than $T = 0.5$. The reason could be that the new node heavily depends on previously generated nodes in the autoregressive generation of $\mathcal{G}^B$. The baseline has more noisy and irrelevant relations between nodes; therefore, it is less efficient to find the scenes that achieve $\mathcal{L}_g$. In addition, a strong exploration makes the irrelevant information have more influence on the baseline. In contrast, our *CausalAF* ignores the insignificant information and focuses on the causation that helps with the goal. We also find that *CausalAF* without CVM performs a little worse than *CausalAF*, which validates our hypothesis that COM may not be powerful enough to represent causality.

6. Conclusion

This paper proposes a causal generative model that generates safety-critical scenes with causal graphs obtained from humans prior. To incorporate the graphical structure of causal graphs, we design a novel scene representation called the Behavioral Graph. The autoregressive generation process of BG makes it possible to inject the causation via regulating the generating order and modifying the graph connection. By introducing causation into generative models, we are able to efficiently create rare scenes that might be difficult to find, such as safety-critical traffic scenes. Our method outperforms the baseline in terms of efficiency and performance on three scenes that have clear causation. One limitation of this work is that the causal graph, usually summarized by humans, is assumed to be always correct. Automatically discovering the causal graph will be the future direction.

References

Abdal, R., Qin, Y., and Wonka, P. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8296–8305, 2020.

Abdelfattah, M., Yuan, K., Wang, Z. J., and Ward, R. Towards universal physical attacks on cascaded camera-lidar 3d object detection models. *arXiv preprint arXiv:2101.10747*, 2021.

Alcorn, M. A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., and Nguyen, A. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4845–4854, 2019.

Arief, M., Huang, Z., Koushik Senthil Kumar, G., Bai, Y., He, S., Ding, W., Lam, H., and Zhao, D. Deep probabilistic accelerated evaluation: A robust certifiable rare-event simulation methodology for black-box safety-critical systems. pp. 595–603, 2021.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Devaranjan, J., Kar, A., and Fidler, S. Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In *European Conference on Computer Vision*, pp. 715–733. Springer, 2020.

Ding, W., Wang, W., and Zhao, D. A new multi-vehicle trajectory generator to simulate vehicle-to-vehicle encounters. *arXiv preprint arXiv:1809.05680*, 2018.

Ding, W., Xu, M., and Zhao, D. Cmts: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4314–4321. IEEE, 2020.

Ding, W., Chen, B., Li, B., Eun, K. J., and Zhao, D. Multimodal safety-critical scenarios generation for decision-making algorithms evaluation. *IEEE Robotics and Automation Letters*, 6(2):1551–1558, 2021a. doi: 10.1109/LRA.2021.3058873.

Ding, W., Li, B., Eun, K. J., and Zhao, D. Semantically controllable scene generation with guidance of explicit knowledge. *arXiv preprint arXiv:2106.04066*, 2021b.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Engel, J., Hoffman, M., and Roberts, A. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*, 2017.

Feng, S., Yan, X., Sun, H., Feng, Y., and Liu, H. X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):1–14, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Grünbaum, A. Causality and the science of human behavior. *American Scientist*, 40(4):665–689, 1952.

Heckerman, D., Geiger, D., and Chickering, D. M. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087. PMLR, 2018.

Jain, L., Chandrasekaran, V., Jang, U., Wu, W., Lee, A., Yan, A., Chen, S., Jha, S., and Seshia, S. A. Analyzing and improving neural networks by generating semantic counterexamples through differentiable rendering. *arXiv preprint arXiv:1910.00727*, 2019.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Khemakhem, I., Monti, R., Leech, R., and Hyvarinen, A. Causal autoregressive flows. In *International Conference on Artificial Intelligence and Statistics*, pp. 3520–3528. PMLR, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kocaoglu, M., Snyder, C., Dimakis, A. G., and Vishwanath, S. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Mollaysa, A., Paige, B., and Kalousis, A. Conditional generation of molecules from disentangled representations. 2019.

Mollaysa, A., Paige, B., and Kalousis, A. Goal-directed generation of discrete structures with conditional generative models. *arXiv preprint arXiv:2010.02311*, 2020.

Najm, W. G., Ranganathan, R., Srinivasan, G., Smith, J. D., Toma, S., Swanson, E., Burgett, A., et al. Description of light-vehicle pre-crash scenarios for safety applications based on vehicle-to-vehicle communications. Technical report, United States. National Highway Traffic Safety Administration, 2013.

O'Kelly, M., Sinha, A., Namkoong, H., Duchi, J., and Tedrake, R. Scalable end-to-end autonomous vehicle testing via rare-event simulation. *arXiv preprint arXiv:1811.00145*, 2018.

Pearl, J. *Causality*. Cambridge university press, 2009.

Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., and Birchfield, S. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255. IEEE, 2019.

Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., and Diermeyer, F. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

Sun, J., Cao, Y., Chen, Q. A., and Mao, Z. M. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 877–894, 2020.

Tan, S., Wong, K., Wang, S., Manivasagam, S., Ren, M., and Urtasun, R. Scenegen: Learning to generate realistic traffic scenes. *arXiv preprint arXiv:2101.06541*, 2021.

Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33, 2020.

Tu, J., Ren, M., Manivasagam, S., Liang, M., Yang, B., Du, R., Cheng, F., and Urtasun, R. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13716–13725, 2020.

Van Ratingen, M., Williams, A., Lie, A., Seeck, A., Castaing, P., Kolke, R., Adriaenssens, G., and Miller, A. The european new car assessment programme: a historical review. *Chinese journal of traumatology*, 19(2):63–69, 2016.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Xiao, C., Yang, D., Li, B., Deng, J., and Liu, M. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6898–6907, 2019.

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. Causalvae: disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9593–9602, 2021.

Zeng, X., Liu, C., Wang, Y.-S., Qiu, W., Xie, L., Tai, Y.-W., Tang, C.-K., and Yuille, A. L. Adversarial attacks beyond the image space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4302–4311, 2019.

Zhao, D., Lam, H., Peng, H., Bao, S., LeBlanc, D. J., Nobukawa, K., and Pan, C. S. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):595–607, 2017. doi: 10.1109/TITS.2016.2582208.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. Dags with no tears: Continuous optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.

Zhu, S., Ng, I., and Chen, Z. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.