Graph Discovery for Visual Test Generation

Neil Hallonquist[1], Donald Geman[2] and Laurent Younes[2]

[1]Applied Physics Laboratory, Johns Hopkins University
[2]Department of Applied Mathematics and Statistics, Johns Hopkins University
neil.hallonquist@yahoo.com geman@jhu.edu laurent.younes@jhu.edu

***Abstract*—We consider the problem of uncovering an unknown attributed graph, where both its edges and vertices are hidden from view, through a sequence of binary questions about it. In order to select questions efficiently, we define a probability distribution over graphs, with randomness not just over edges, but over vertices as well. We then sequentially select questions so as to: (1) minimize the expected entropy of the random graph, given the answers to the previous questions in the sequence; and (2) instantiate the vertices that compose the graph. We propose some basic question spaces, from which to select questions, that vary in their capacity.**

We apply this framework to the problem of test generation in Visual Question Answering (VQA), where semantic questions are used to evaluate vision systems over rich image representations. To do this, we use a restricted question vocabulary, resulting in image representations that take the form of scene graphs; by defining a distribution over them, a consistent set of probabilities is associated with the questions, and used in their selection.

I. Introduction

There are often situations in which there is an unknown (attributed) graph, one that is not directly observable, that needs to be either fully or partially determined. This is a common occurrence since, in general, entities in the world are not directly observable, but must be inferred from some signal. In this work, we consider the active refinement of knowledge about such graphs through sequential questioning, which we refer to as *graph discovery*. We consider the case in which the vertices, in addition to the edges, are unknown, which introduces the issue of vertex instantiation, the search for some characteristic set of attributes that distinguishes a vertex from all others in the graph, allowing its labeling and identification. Once a vertex is instantiated, more specific questions can be posed to further discern its attributes and edges incident. Graph discovery can be seen as an instance of entity identification [1], [2], [3], [4], [5], tailored to the specific structure of graphs.

Suppose we have a graph space $\mathcal{G}$, containing graphs that vary in their vertices, and let questions be functions of the form $f : \mathcal{G} \to \{0, 1\}$, taking graphs to their binary answer. Given a distribution over $\mathcal{G}$, a natural criterion for selecting questions is the minimization of the expected entropy of the random graph, which in turn, corresponds to the selection of maximum entropy questions, i.e., those in which the answer cannot be reliably guessed. To form question sequences, distributions are conditioned on the previous questions and answers in the sequence. For defining distributions over the graph space $\mathcal{G}$, we use an exponential random graph model, and we give an extension of the decomposition theorem for Gibbs distribution to it.

The question space from which questions are selected is important, dictating the extent to which graphs can be determined as well as how efficiently. We propose several types of questions, including coarse ones about the existence of vertices, and finer ones about their uniqueness, allowing instantiations using coarse-to-fine search [6]. We propose questions that qualify these existence inquiries to only those vertices that have not yet been instantiated, which increases the size of the question space and its resolution capacity.

We apply this framework to visual test generation using semantic questions, a problem in the field of Visual Question Answering (VQA) [7], [8], [9], [10], [11], [12], [13], [14], [15]. To comprehend imagery, vision systems must be able to produce correct, compressed representations of them, and under restrictions on their semantic content, these representations take the form of *scene graphs* [16], [17], [18], [19], [20]. In these graphs, vertices represent objects (e.g., people and vehicles), and edges represent relationships between objects (e.g., holding hands, driving, etc.), and further, they do not have constant order (e.g., a scene may be empty or may have numerous objects). Thus, images, before they are interpreted, correspond to unknown graphs, and for formulating tests, we make the assumption that questions that are informative about these underlying graphs are also informative for vision system evaluation. Visual test generation is related to *visual question generation* [21], [22], [23], [24], extended to question sequences. Questions that instantiate objects are *referring expressions* [25], [26], [27], [28], [29], [30].

II. Graph Discovery

In this section, we formalize our approach to graph discovery, defining the graph and question spaces of interest, and discuss question predictability and vertex instantiation. In the next section, we consider an exponential model for graph spaces where graphs can vary in their vertices, then consider methods for conditional sampling from a distribution over this space, conditioned on a sequence of questions and answers.

A. Attributed Graphs

To define attributed graphs, the following components are required.

(i) A set Λ_V of *vertex attributes*.
(ii) A set Λ_E of *edge attributes*.

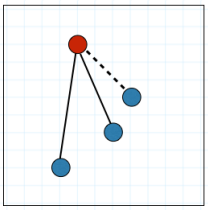


Fig. 1: A grid-supported graph associated with the toy example, where there are two types of edge attributes represented by solid and broken lines.

(iii) A set K_V of *vertex identifiers* together with a surjection $\kappa : \Lambda_V \to K_V$.
(iv) An element $0_E \in \Lambda_E$ used to designate the absence of an edge.

For simplicity, we assume that Λ_V and Λ_E are finite.

Definition II.1. An (attributed) graph is a pair $G = (V, E)$ where

(1) V (the set of vertices) is a subset of Λ_V such that the restriction of κ to V is one-to-one.
(2) E (representing the edges) is a mapping $E : \Lambda_V \times \Lambda_V \to \Lambda_E$ such that $E(v, v') = 0_E$ if $v \notin V$ or $v' \notin V$

The identity $E(v, v') = 0_E$ means that there is no edge from v to v' (with $v, v' \in V$). We will denote the empty graph (for which $V = \emptyset$) by $G = \emptyset$. We let $\mathcal{G}(\Lambda_V, \Lambda_E)$ (or simply $\mathcal{G}$ when Λ_V and Λ_E are clear from the context) denote the space of all graphs associated with Λ_V and Λ_E. The condition on the restriction of κ to V ensures that no pair of vertices share the same identifier.

With this definition, a graph has no loop if and only if $E(v, v) = 0_E$ for all $v \in V$ and it is undirected if and only if $E(v, v') = E(v', v)$ for all $v, v' \in V$.

Running Example. Let the vertex space be $\Lambda_V = \{1, \ldots, p\} \times \{\text{blue}, \text{red}\}$ where $1, \ldots, p$ are location labels (e.g., the pixels in an image). Taking $\kappa(i, c) = i$ and $\Lambda_E = \{0_E\}$ models a set of grid points each labeled by a unique color (yielding a marked point process). A non-trivial edge space would allow for having different types of interactions between the vertices, for example, $\Lambda_E = \{0_E, \text{solid}, \text{broken}\}$ where a probability distribution on $\mathcal{G}$ could allow for the likelihood an an edge to depend on the colors at the two corresponding vertices (see Fig. 1).

B. Question Predictability

Let a question be a function of the form $f : \mathcal{G} \to \{0, 1\}$, mapping a graph from $\mathcal{G}$ to a binary answer. Suppose we have asked a sequence of questions $F_k = (f_1, \ldots, f_k)$; the answers to these questions on a graph G are

$$A_k = F_k(G) = (f_1(G), \ldots, f_k(G)).$$

Let (Ω, P) be a probability space and $\mathbf{G} : \Omega \to \mathcal{G}$ be a graph-valued random variable (a random graph). Define the conditional probability of a question f having a positive answer given a history $H = (F_k, A_k)$ as:

$$P_H(f(\mathbf{G}) = 1) \equiv P(f(\mathbf{G}) = 1 \mid F_k(\mathbf{G}) = A_k). \quad (1)$$

The *predictability* of a question f with respect to history H is then defined as

$$\rho(f, H) = |P_H(f(\mathbf{G}) = 1) - 0.5| \,. \quad (2)$$

A question f is *unpredictable* if $\rho(f, H) = 0$ and ϵ-unpredictable if $\rho(f, H) \leq \epsilon$. We call these questions unpredictable because, without explicit knowledge of the graph, they cannot be reliably guessed given the answers to the previous questions.

Remark II.1 (Entropy Minimization). Unpredictable questions are desirable from an information-theoretic perspective: they minimize the expected entropy of a random graph. Let f be an unpredictable question with respect to history $H = (F, A)$, and define the event $\mathcal{G}_H \equiv \{G \in \mathcal{G} \mid F(G) = A\}$. Then, $P_H(f(\mathbf{G}) = 1) = 0.5$ is equivalent to $\mathcal{H}(f(\mathbf{G}) \mid \mathcal{G}_H) = 1$ where $\mathcal{H}$ is the entropy and conditioning on $\mathcal{G}_H$ means taking the entropy for the conditional distribution of $\mathbf{G}$ given $\{\mathbf{G} \in \mathcal{G}_H\}$. This implies that

$$f = \arg\min_{f' \in \mathcal{F}} \mathcal{H}(\mathbf{G} \mid \mathcal{G}_H, f'(\mathbf{G}))$$

where

$$\mathcal{H}(\mathbf{G} \mid \mathcal{G}_H, f(\mathbf{G})) = \sum_{a \in \{0,1\}} P(f(\mathbf{G}) = a \mid \mathcal{G}_H)\, \mathcal{H}(\mathbf{G} \mid \mathcal{G}_H, f(\mathbf{G}) = a).$$

The above implication follows because

$$\mathcal{H}(\mathbf{G} \mid \mathcal{G}_H, f(\mathbf{G})) = \mathcal{H}(\mathbf{G}, f(\mathbf{G}) \mid \mathcal{G}_H) - \mathcal{H}(f(\mathbf{G}) \mid \mathcal{G}_H)$$

and, since the question f is a function of a graph, we have

$$\mathcal{H}(\mathbf{G}, f(\mathbf{G}) \mid \mathcal{G}_H) = \mathcal{H}(\mathbf{G} \mid \mathcal{G}_H).$$

Suppose we have some question space $\mathcal{F}$ from which to select questions. To form a question sequence for a given unknown graph, the next question in the sequence is chosen by: (1) estimating the predictability of each question in $\mathcal{F}$; and (2) selecting a question from among those that are ϵ-unpredictable:

$$\mathcal{U}_H \equiv \{f \in \mathcal{F} \mid \rho(f, H) \leq \epsilon\},$$

where H is the current history. The predictabilities can be estimated by sampling from the conditional distribution P_H (see section IV).

C. Questions

In this section, we describe some useful questions, first for instantiating vertices, then for refining our knowledge about them.

Simple Questions. Let Λ_V be the vertex space (associated with the graph space $\mathcal{G}$) and let $B \subset \Lambda_V$. For $G = (V, E)$, let

$N_B(G) = |V \cap B|$ be the number of vertices in V that belong to B. We define questions that can establish the existence of vertices in the unknown graph.

1. An *existence question* has the form:

$$f_B(G) = I_{\{N_B(G)>0\}} \text{ where } B \subset \Lambda_V.$$

It has a positive answer if the graph G has one or more of its vertices in B. Let the set of all existence questions be denoted by $\tilde{\mathcal{F}}_{\text{exist}}$.

2. A *uniqueness question* has the form:

$$f_B(G) = I_{\{N_B(G)=1\}} \text{ where } B \subset \Lambda_V.$$

It has a positive answer if the graph G has one and only one of its vertices in B. Notice that for any uniqueness question f_B, if a graph G receives a positive answer (i.e. $f_B(G) = 1$), a vertex in this graph can be uniquely identified through the set B, which distinguishes this vertex from all others in the graph. We consider the idea of vertex instantiation in the next section. Let the set of all uniqueness questions be denoted by $\tilde{\mathcal{F}}_{\text{uniq}}$.

Running Example (cont) Suppose the graph G shown in Fig. 2 is an unknown graph that we want to determine. Let $B \subset \Lambda_V$ be the subset containing all vertices that have a location component in the orange rectangle depicted in the left panel of figure (i.e., $B = \{v \in \Lambda_V \mid \text{loc}(v) \in R\}$, where $R \subset \mathcal{L}$ is the subset of locations depicted by the rectangle, and $\text{loc}(v)$ denotes the projection of v to its location component). Then, for an existence question f_B, we have that $f_B(G) = I_{\{N_B(G)>0\}} = 1$ since the graph has a vertex in B, and for a uniqueness question f_B, we have that $f_B(G) = I_{\{N_B(G)=1\}} = 1$ since the graph has one and only one vertex in B. Thus, this latter question uniquely identifies this vertex from the others in the graph.

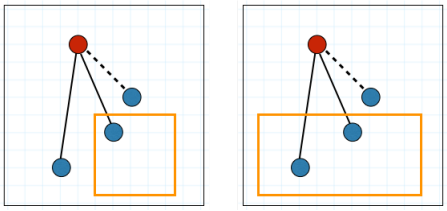


Fig. 2: The designated region in the left panel can be used to instantiate the vertex in it. The designated region in the right panel can also be used to instantiate a vertex, given that one of the two vertices in the region is already instantiated.

We define the set of instantiation questions as $\tilde{\mathcal{F}}_{\text{inst}} \equiv \tilde{\mathcal{F}}_{\text{exist}} \cup \tilde{\mathcal{F}}_{\text{uniq}}$. In practice, there are limits on the set of questions that can be used; for example, instantiation questions may be uninterpretable for many $B \subset \Lambda_V$ (uninterpretable in the sense that there does not exist a compact description of the question based on the available vocabulary). Thus, assume we only use instantiation questions that involve $B \in \mathcal{B}$, where $\mathcal{B} \subset \mathbb{P}(\Lambda_V)$ is some collection of allowable sets. We denote this set of questions by:

$$\tilde{\mathcal{F}}_{\text{inst}}(\mathcal{B}) \equiv \{f_B \in \tilde{\mathcal{F}}_{\text{inst}} \mid B \in \mathcal{B}\}.$$

Instantiation. Instantiation of a vertex refers to: (1) the discovery of some set $B \subset \Lambda_V$ that distinguishes that vertex from all others in the unknown graph; and (2) the denotation of this unique vertex with a label (e.g. vertex v_i). Let $G = (V, E)$ be an unknown graph and suppose we have asked a sequence of instantiation questions $F_k = (f_1, \ldots, f_k)$ about it. Denote the set of vertices instantiated by these questions by $\Omega^{\text{Inst}}(G, F_k) \subset V$ and denote the set of vertices that have not been instantiated by $\Omega^{\neg\text{Inst}}(G, F_k)$. Furthermore, assume the set $\Omega^{\text{Inst}}(G, F_k)$ is ordered according to instantiation time, writing it in the form:

$$\Omega^{\text{Inst}}(G, F_k) = (v_1, \ldots, v_{n'}),$$

where each $v_i \in V$ and the index indicates this vertex was the ith vertex to be instantiated. Although a vertex $v \in \Omega^{\text{Inst}}(G, F_k)$ has been instantiated, in general, it is not completely known: we only know that it exists in some subset $B \subset \Lambda_V$.

Because every vertex in a graph is unique (by our definition of a graph, $V \subset \Lambda_V$ does not allow for duplicate vertices), every vertex can be instantiated if one is allowed to ask questions associated to any subset $B \subset \Lambda_V$. This would be however impractical, because the space of possible questions to explore at each step of the algorithm would be intractably large. When working with a coarse space of possible set B, it is possible that there is no available questions capable of instantiating a given vertex v of the unknown graph because $N_B \geq 2$ for any set B used in the questions and containing v. Such an event can be made considerably less likely by progressively removing instantiated vertices from consideration, which we now discuss.

Complex Questions. We now consider a larger space of questions, one with questions that can be directly applied to unexplored regions of an unknown graph. These questions can: (1) refine our knowledge of vertices that have already been instantiated, as well as the edges between them; or (2) focus on discovering new vertices that have not been instantiated.

1) Post-Instantiation Questions: Suppose we have a graph G and a sequence of questions $F_k = (f_1, \ldots, f_k)$, and have instantiated a set of vertices:

$$V' = \Omega^{\text{Inst}}(G, F_k) = (v_1, \ldots, v_{n'}).$$

As mentioned above, we only have a partial knowledge of these instantiated vertices. Suppose we want to learn more about them. This may be done by focusing attention on the subgraph $G' = G(V')$, i.e. the graph induced by the instantiated vertices. Let's consider some examples of post-instantiation questions.

1. Vertex refinement questions: Often, we want to focus attention on a single vertex, say $v_i \in V'$, the ith instantiated

vertex. A question $f_{B,F_k,i}$ is a *vertex refinement question* if it has the following form: $f_{B,F_k,i}(G) = I_{\{v_i \in B\}}$. Since we already have a partial knowledge of the vertex v_i, for example, that it exists in $B' \subset \Lambda_V$, we will want to ask vertex refinement questions in which $B \subset B'$.

2. Edge refinement questions: Suppose we want to learn about the edge value $E(v_i, v_j)$ between $v_i, v_j \in V'$, the ith and j^{th} instantiated vertices. An edge refinement question can be formulated as $f_{B,F_k,(i,j)} = I_{\{E(v_i,v_j) \in B\}}$.

Relative Instantiation Questions. In addition to focusing attention on instantiated vertices, we also want to gather information about non-instantiated ones and in particular quickly instantiate new vertices. This can be done more efficiently by removing from consideration vertices that have already been instantiated. Suppose we have asked a sequence of questions $F_k = (f_1, \ldots, f_k)$ and we want to gain information about the non-instantiated vertices:

$$V' = \Omega^{\neg\text{Inst}}(G, F_k) \subset V.$$

This may be done by forming a new graph $G' = G(V')$, i.e. the graph induced by the non-instantiated vertices. The existence and uniqueness questions that were defined previously had the form $f_B(G) = I_{\{N_B(G)>0\}}$ or $I_{\{N_B(G)=1\}}$, where $B \subset \Lambda_V$.

We now introduce history-dependent existence and uniqueness questions of the form: $f_{B,F_k}(G) = I_{\{N_B(G')>0\}}$ and $I_{\{N_B(G')=1\}}$ where $G' = G(V')$ and $V' = \Omega^{\neg\text{Inst}}(G, F_k)$. We will denote this new set of existence and uniqueness questions by $\mathcal{F}_{\text{exist}}$ and $\mathcal{F}_{\text{uniq}}$, respectively, and let $\mathcal{F}_{\text{inst}} \equiv \mathcal{F}_{\text{exist}} \cup \mathcal{F}_{\text{uniq}}$. As above, we let $\mathcal{F}_{\text{inst}}(\mathcal{B})$ denote the set of instantiation questions using $B \in \mathcal{B}$, where $\mathcal{B} \subset \mathbb{P}(\Lambda_V)$ is some collection of allowable sets. We will refer to these questions as relative instantiation, because they refer to information contained in the history.

Running example (cont.) Suppose we ask a sequence of questions F_k containing the uniqueness question f_B, where $B \subset \Lambda_V$ is the set of vertices contained in the orange rectangle on the left panel of Fig. 2. Then, this sequence instantiates the vertex v_1 in this rectangle (i.e., we have $v_1 \in \Omega^{\text{Inst}}(G, F_k)$). Suppose this is the only vertex instantiated by the sequence, and to instantiate more vertices, we ask another uniqueness question. Let $C \subset \Lambda_V$ be the set of vertices contained in the orange rectangle on the right panel of Fig. 2. Then, for the uniqueness question $f_{C,F_k} \in \mathcal{F}_{\text{uniq}}$, we have that $f_{C,F_k}(G) = I_{\{N_C(G')=1\}} = 1$, and we can instantiate the vertex to the left of vertex v_1. Notice, however, if we instead ask the uniqueness question $f_C \in \tilde{\mathcal{F}}_{\text{uniq}}$, then we have that $f_C(G) = I_{\{N_C(G)=1\}} = 0$, and this other vertex is not instantiated.

Relative instantiation questions have at least two advantages relative to "absolute" ones. The first advantage addresses the uniqueness of instantiations; when using questions in $\tilde{\mathcal{F}}_{\text{inst}}$, it is possible to instantiate the same vertex multiple times: whenever a uniqueness question $f_B \in \tilde{\mathcal{F}}_{\text{uniq}}$ receives a positive answer, there is no way to know whether the unique vertex in B has already been instantiated before or whether it is a new discovery. Removing previously instantiated vertices from those that are admissible is therefore essential to address this ambiguity.

The second difference concerns the capacity to interpret the graph space. As we remarked if the set $\mathcal{B}$ offers a resolution which is too coarse compared to the vertices in the graph, a large number of questions will not allow one to identify a unique vertex. When one progressively removes instantiated vertices from consideration, more uniqueness questions become true allowing for a more efficient exploration of the graph space. This increased capacity comes at a cost however: the estimation of statistics for relative instantiation questions tends to require more data.

III. Probability Distributions on Graphs

We will use maximum entropy graph models (also called exponential random graphs models) to define probability distributions on attributed graphs. We fix Λ_V, κ and Λ_E and let $\mathcal{G} = \mathcal{G}(\Lambda_V, \kappa, \Lambda_E)$. Given a family of sufficient statistics $U_1, \ldots, U_K$, with $U_k : \mathcal{G} \to \mathbb{R}$, and real numbers $u_1, \ldots, u_K$, the probability distribution that has maximum entropy subject to constraints $\mathbb{E}(U_k) = u_k$, $k = 1, \ldots, K$ (where $\mathbb{E}$ refers to expectation) must, if it exists, take the form

$$P_\lambda(G) = \frac{1}{Z_\lambda} \exp\left(-\sum_{k=1}^{K} \lambda_k U_k(G)\right). \tag{3}$$

(Conditions for the existence of the maximum entropy distribution require that the set of probability distributions satisfying the constraints is not empty and that $(u_1, \ldots, u_K)$ lies within the relative interior of the set of feasible constraints.) The optimal $\lambda = (\lambda_1, \ldots, \lambda_K)$ can be obtained by solving the system of equations

$$\mathbb{E}_\lambda(U_k) = u_k,\ k = 1, \ldots, K \tag{4}$$

where $\mathbb{E}_\lambda$ denotes the expectation for P_λ.

We give a theoretical result on random graphs that generalizes the proof of the existence of a potential associated to a Gibbs distribution on a product space. Theorem 1 below is (slightly) more general than decomposition theorems on exponential random graphs such as those studied in [31], [32], in that it addresses at once the randomness in the number of vertices and in the edge set. We start with the following definition.

Definition III.1. Let $G = (V, E), G' = (V', E') \in \mathcal{G}$. One says that G' is a subgraph of G (with notation $G' \preceq G$) if $V' \subset V$ and $E'(v, v') \in \{E(v, v'), 0_E\}$ for all $(v, v') \in V' \times V'$. We will also use the notation $E' \preceq E$ to describe the preceding condition.

In the special case in which $E'(v, v') = E(v, v')$ if $v, v' \in V'$ and 0_E otherwise, one says that G' is the restriction of G to V' and write $G' = G_{V'}$ (or $E' = E_{V'}$).

Theorem 1. *Every positive probability π on $\mathcal{G}$ can be expressed in the form*

$$\pi(G) = \frac{1}{Z} \exp\left(- \sum_{G' \preceq G} \Phi(G') \right). \tag{5}$$

for some constant Z and a function $\Phi : \mathcal{G} \to \mathbb{R}$. Moreover, such a Φ is unique subject to the condition $\Phi(\emptyset) = 0$. (Proof in supplement).

This generalization of the decomposition theorem for Gibbs distributions over product spaces makes possible the definition of probability distributions on $\mathcal{G}$ by specifying the value of Φ on small graphs in $\mathcal{G}$. Note that, if the sets of vertex or edge attributes are large, even the definition of $\Phi(G)$ for small G can have high complexity. One can easily extend the previous theorem to also decompose configurations of attributes into sums of functions of increasing complexity, even though we will not explicitly state this notation-heavy generalization here.

IV. Conditional Sampling

Let $\mathcal{G}$ denote a finite graph space and let P be a distribution over it. Suppose we have a history $H = (F, A)$, and let $\mathcal{G}_H$ be the set of graphs that coheres with this history:

$$\mathcal{G}_H \equiv \{G \in \mathcal{G} \mid F(G) = A\}. \tag{6}$$

To sample from conditional distributions of the form $P_H(G) \equiv P(G \,|\, \mathcal{G}_H)$, the Metropolis-Hastings algorithm can be used, with transition probability q_H designed to be easily sampled from (e.g., a uniform distribution on a finite set of elementary moves). In this work, we use the following elementary moves: vertex addition, vertex deletion, and individual vertex or edge modification (details in supplement). This will generate a sequence $G_1, G_2, \ldots$ of graphs. To ensure that this chain has the stationary distribution P_H, it suffices for it to be ergodic. Since the graph space is finite, a chain is ergodic if there exists a number n such that the chain can go from any graph to any other graph in exactly n steps with positive probability. We can guarantee ergodicity by checking two sufficient conditions: (1) the chain can reach any graph in the set $\mathcal{G}_H$ from any other graph in this set with positive probability; and (2) for any graph in this set, there is a positive probability of returning to itself in one step.

Notice that the satisfaction of these requirements, and in turn, the difficulty of conditional sampling, depends on the questions in the history H. If questions can be arbitrary (i.e., questions can be any function $f : \mathcal{G} \to \{0, 1\}$), then the space $\mathcal{G}_H$ can be convoluted and disconnected with respect to the simple, first-order moves employed. Indeed, with arbitrary questions, the space $\mathcal{G}_H$ can be an arbitrary set, and hence conditional sampling would be almost impossible without a brute force approach. Thus, we restrict our attention to the questions described in the previous section. In particular, we only consider instantiation questions in $\mathcal{F}_{\text{inst}}$ rather than in $\tilde{\mathcal{F}}_{\text{inst}}$; these questions have the valuable property that vertices can only be instantiated once. Notice that if a history has questions from $\tilde{\mathcal{F}}_{\text{inst}}$, then the space $\mathcal{G}_H$ can be disconnected with respect to first-order moves. For example, suppose a sequence has two uniqueness questions $f_{B_1}, f_{B_2} \in \tilde{\mathcal{G}}_{\text{uniq}}$, where $B_1 \neq B_2$ and $B_1 \cap B_2 \neq \emptyset$, and both questions have positive answers. Then, the unknown graph either has one vertex in $B_1 \cap B_2$, or two vertices, one in $B_1 \setminus (B_1 \cap B_2)$ and the other in $B_2 \setminus (B_1 \cap B_2)$. This is a problem because there is no way to transition between these two scenarios with moves where a proposed graph can only differ from the current graph by one vertex.

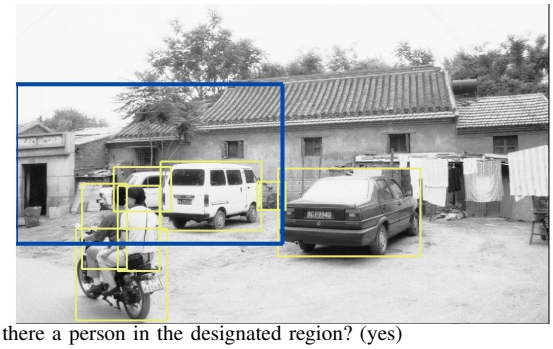
V. Experiments

For an application of this framework, we use it to evaluate the scene understanding of vision systems, machines designed to take imagery (or video) and produce compressed representations of it. Traditionally, to evaluate them, image representations are completely recorded by humans; as they increase in complexity though, this approach becomes infeasible and testing can only be partial. In VQA, the representation space is implicitly defined in terms of semantic questions, and given an image, parts or aspects of its representation can be tested using them, allowing evaluations to scale. In this work, by using a restricted vocabulary, the image representations take the form of scene graphs, and we generate tests in the form of binary question sequences.

For tests to be efficient, the questions cannot be selected arbitrarily; some binary questions, for example, often can be answered correctly using only information external to the image (e.g., using "language priors" or statistics of the image population), as pointed out in [15]. To illustrate this issue, Fig. S.IV in the supplement gives a histogram of question predictability given a history of ten previous queries; it shows that most questions are highly predictable (i.e., could be answered reliably based on prior knowledge), and unpredictable questions would almost never be obtained through random sampling. In [15], this testing issue was addressed by, for a given question, finding two similar images with opposite answers. Here, in contrast, we will take the image space (and distribution over it) as fixed, and select questions that are unpredictable with respect to it. A benefit of this approach is that test (question sequence) error rates can be used as estimates of test performance in the field, under traditional statistical learning assumptions.

Scene Graphs. In our experiments, we use the image dataset from [33], which contains 2,591 images of street scenes, annotated as scene graphs with the following components. Let $\mathcal{T} = \{\text{person}, \text{vehicle}\}$ be a space of object types, let $\mathcal{W}$ be a set of rectangles assigning a position and a scale to each object, and let $\mathcal{A}_t$ be the set of traits that can be attached to an object of type $t \in \mathcal{T}$. Hence, every object will be assigned a type $t \in \mathcal{T}$, a bounding box $w \in \mathcal{W}$ and a type-dependent trait $a \in \mathcal{A}_t$, and will therefore be represented by a vertex $v = (t, w, a)$ in the set

$$\Lambda_V = \{(t, w, a) \mid t \in \mathcal{T}, w \in \mathcal{W}, a \in \mathcal{A}_t\}.$$



- Is there a person in the designated region? (yes)
- Is there a unique person that is an adult in the designated region? (no)
- Is there a unique person that is carrying something in the designated region? (yes: person 1)

Fig. 3: A selection of questions from a much longer sequence provided in the supplement. The last question makes possible the instantiation of person 1.

In our setting, the sets $\mathcal{A}_t$ are, in addition, defined as product spaces $\mathcal{A}_t^1 \times \cdots \times \mathcal{A}_t^{n(t)}$ where $\mathcal{A}_t^j$ defines a (small) set of mutually exclusive properties (e.g., child vs. adult; or sitting vs. standing, etc.). We define $\kappa(t, w, a) = (t, w)$, ensuring that no more that one object of a given type can be found at a given location. (Details of these sets are in tables S.I and S.II).

To define the edge space, we let $\mathcal{R}$ denote the set of all possible types of relationships between interacting objects. Then $\Lambda_E = \mathcal{P}(\mathcal{R})$ is the family of all subsets of $\mathcal{R}$, so that $0_E = \emptyset$ represents an absence of interaction. Note that two objects may have more than one type of interaction (e.g., holding hands while talking). Associated to a pair of types t, t', we also assume that a subset $\mathcal{R}_{t,t'} \subset \mathcal{R}$ specifies a family of allowed relationships These are handled in the stochastic model by ensuring that forbidden interactions have probability 0.

Statistical Model. To define a distribution over the scene graph space, we make a series of simplifying assumptions. For a vertex $v = (t, w, a) \in \Lambda_V$, we will write $T(v) = t$, $W(v) = w$ and $A(v) = a$. Letting $\delta(e) = I_{\{e \neq 0_E\}}$, we define the transformation $G = (V, E) \mapsto \sigma(G) = (\kappa(V), \delta(E))$, mapping $\mathcal{G}$ to $\mathcal{G}' = \mathcal{G}(\mathcal{T} \times \mathcal{W}, \kappa', \{0, 1\})$ (with $\kappa'(t, w) = (t, w)$). We will refer to $\sigma(G)$ as the "skeleton" of G. We define a probability distribution P on $\mathcal{G}$ corresponding to a two-step generative process - choosing a skeleton and then fleshing it out with traits. We therefore decompose

$$P(G) = Q(G \,|\, \sigma(G)) \cdot \mu_0(\sigma(G)), G \in \mathcal{G}, \tag{7}$$

where μ_0 is a probability distribution on $\mathcal{G}'$, that we will model as a maximal entropy graph distribution given in equation (3). The features $U_1, \ldots U_K$ in the model consist of $K = 21$ simple, hand-crafted functions that describe the spatial configuration and relationships of objects (e.g., the number of objects at a given location and their sizes); see the supplement for details. Some examples of the dataset and some samples from the model μ_0 are shown in Fig. 4. For the parameters of μ_0, we use the maximum likelihood estimate, solved for using a stochastic optimization algorithm [34], [35], [36], [37].

We assume that the remaining components $(A(v), v \in V)$ and $(E(v, v'),\ v, v' \in V)$ are mutually conditionally independent given $\kappa(V)$ and $\delta(E)$. Moreover, we assume that the simple traits composing $A(v)$ are also independent and independent of the rest of the variables. We therefore introduce probability distributions $\varphi_t^{(j)}$ on $\mathcal{A}_t^{(j)}$ for $t \in \mathcal{T}$ and $j = 1, \ldots, n(t)$, and $\varphi_{t,t'}$ on $\mathcal{R}_{t,t'}$ for $t, t' \in \mathcal{T}$ such that, for $G = (V, E) \in \mathcal{G}$

$$Q(G \,|\, \sigma(G)) = \prod_{v \in V} \prod_{j=1}^{n(T(v))} \varphi_{T(v)}^{(j)}(A^{(j)}(v)) \prod_{v, v' \in V} \varphi_{T(v), T(v')}(E(v, v')). \tag{8}$$

where the jth component of $A(v)$ will be denoted $A^{(j)}(v)$. Notice that Q, as a product of univariate probabilities, is the sum of all $Q(G|\sigma(G))$ over all graphs that share the same skeleton is 1. This implies in turn that the function P defined in (7) is a probability distribution on $\mathcal{G}$.

Test Generation. For the questions space, we use the relative existence and uniqueness questions in $\mathcal{F}_{\text{inst}}$ for object instantiation, as well as the post-instantiation questions for vertex and edge refinement (see section II-C). We limit questions to those with compact semantic descriptions. To generate tests, ϵ-unpredictable questions are sequentially selected, using $\epsilon = 0.15$. On each iteration, question predictabilities are estimated by sampling from the conditional distribution; this sampling requires initial states (i.e., starting graphs) that cohere with the history H, which are obtained by using the samples from the previous iteration that cohere with it (roughly 50% of them by construction). Details are shown in the supplement.

A. Validation Study

1) Validation of the sampling algorithm: To check the accuracy of our conditional sampling algorithm, the following experiment was conducted. Given a history H_k of length $k = 10$, we form two datasets of graphs as follows.

(1) The first dataset $\mathcal{G}_0$ is formed by sampling according to the distribution P_{H_k} using the conditional sampling algorithm in the previous section.
(2) The second dataset $\mathcal{G}_1$ is formed by: (a) sampling a million graphs according to the distribution P; and (b) filtering these samples to keep only those that cohere with the history H_k.

Assuming that H_k is a sequence of unpredictable questions, the probability that a randomly sampled graph coheres with it is about 2^{-k}, so that, after 10 steps, one can expect that about a thousand out of the million sampled graphs in the second dataset will be kept in $\mathcal{G}_1$. We generated the same number of simulated samples to form $\mathcal{G}_0$.

By design, the dataset $\mathcal{G}_1$ is distributed according to the true conditional distribution. To test the validity of the conditional

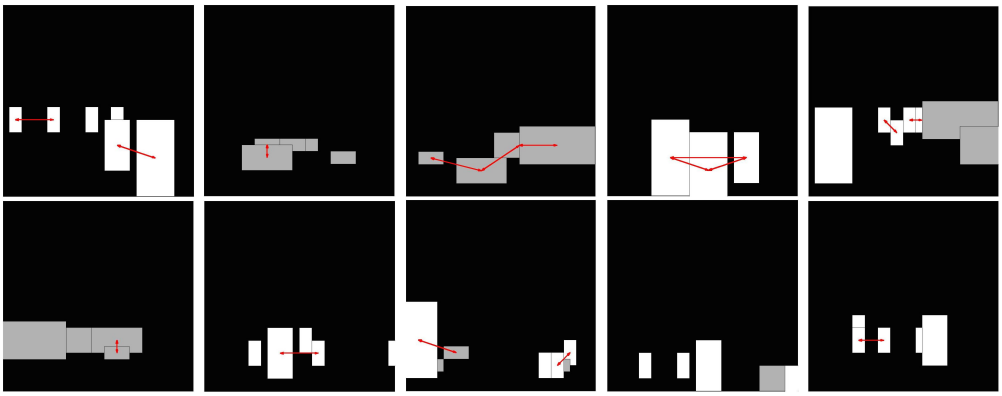


Fig. 4: *Top row*: Examples from dataset used to learn the skeleton-graph model μ_0. *Bottom row*: Samples from skeleton-graph model μ_0.

sampling algorithm, we compare $\mathcal{G}_0$ to $\mathcal{G}_1$ by comparing the histograms for the likelihoods of the samples (see supplementary material, Fig. S.V) and comparing the mean value of scene features in the datasets (table S.III). For example, the mean value for feature 1 using $\mathcal{G}_1$ (resp., $\mathcal{G}_0$) is 2.030 (resp., 2.315), for feature 2 is 1.341 (resp., 1.418), for feature 3 is 0.0062 (resp., 0.00154). We also compared the percentage of objects in each dataset with various trait values; for example, the percentage of males with $\mathcal{G}_1$ (resp., $\mathcal{G}_0$) is 0.4386 (resp. 0.4601), the percentage of persons standing still is 0.4241 (resp. 0.4002); for interaction, the percentage of persons talking is 0.5838 (resp. 0.5740), etc. (Details in tables S.IV and S.V). Although the conditional sampling algorithm should work in theory, this analysis indicates it also works in practice (or, to be more precise, doesn't show any obvious signs of being amiss). Examples of conditional samples are shown in Fig. 5.

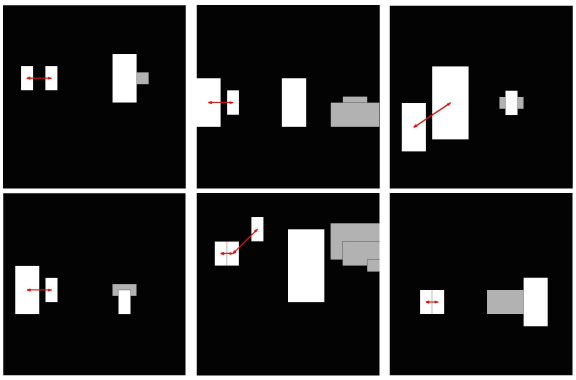


Fig. 5: Conditional samples from the model, given the history $H_k = (F_k, A_k)$ shown in supplement (sequence 3). Loosely, this history instantiates two people on the left-half of the image that are interacting with each other, instantiates a person on the right-half, and instantiates a vehicle on the right-half.

2) Model Validation: We test the validity of our scene model with respect to the task of selecting unpredictable questions. Suppose we have a graph G and we generate a sequence of questions $(f_1, \ldots, f_l)$ for it using our statistical model. For each selected question f_k, we will record the estimated conditional probability of that question having a positive answer, as well as its actual answer $a_k = f_k(G)$. If we collect this data over enough (randomly selected) graphs, then by comparing the estimated number of positive answers to the observed number, we can assess the accuracy of the model's conditional probability estimates, and in turn, the accuracy of the model in selecting ϵ-unpredictable questions.

Applying this, we randomly sampled 50 images from the training set and, for each of them, generated a sequence of length $k = 25$. Out of the 1250 questions generated this way, the expected percentage of yes answers was 47.1%, while the observed one was 41.0%. Table I shows the prediction accuracy as a function of the position in the sequence and of the type of question. This table indicates that the predictability of questions according to the model often differs from their true predictability, yet on average, the model selects questions that fall within the acceptable range for being considered unpredictable, i.e. on average, the questions are unpredictable within a tolerance of $\epsilon = 0.15$.

	Estimated #yes	Observed #yes
Questions 1-5	121.7 (48.7%)	94 (37.6%)
Questions 6-10	115.1 (46.0%)	94 (37.6%)
Questions 11-15	114.6 (45.8%)	100 (40.0%)
Questions 16-20	115.2 (46.1%)	103 (41.2%)
Questions 21-25	121.1 (48.5%)	121 (48.4%)
Instantiation questions	363.1 (48.2%)	282 (37.5%)
Trait questions	195.2 (45.5%)	200 (46.6%)
Relationship questions	29.4 (43.3%)	30 (44.1%)

TABLE I: Accuracy of the model in producing unpredictable questions based on the question's position in the sequence or the question's type (50 images, 25 questions per image).

VI. Discussion

We considered the active refinement of knowledge about unknown graphs using binary questions, where both vertices and edges are hidden from view. These ideas were applied to visual test generation, where we learned a distribution over scene graphs, allowing consistent probabilities to be associated with semantic questions, and the sequential selection of unpredictable ones. These questions ensure tested systems

cannot only use information external to the image. Importantly, however, question predictability is not the same as question difficulty. Questions about a dominant color, for example, can be easier than those about a dominant texture, even when both are equally unpredictable. Difficulty information may be derived by studying the performance of a population of systems. This information might permit adaptive testing [38], [39] on vision systems, where tests adapt based on the answers given and focus on where performance estimates are most uncertain, a possible direction of future research.

Acknowledgment. This work was partially supported by the ONR grant N00141512267.

REFERENCES

[1] M. R. Garey, "Optimal binary identification procedures," *SIAM Journal on Applied Mathematics*, vol. 23, no. 2, pp. 173–186, 1972.

[2] M. R. Garey and R. L. Graham, "Performance bounds on the splitting algorithm for binary testing," *Acta Informatica*, vol. 3, no. 4, pp. 347–355, 1974.

[3] D. W. Loveland, "Performance bounds for binary testing with arbitrary weights," *Acta Informatica*, vol. 22, no. 1, pp. 101–114, 1985.

[4] B. M. Moret, "Decision trees and diagrams," *ACM Computing Surveys (CSUR)*, vol. 14, no. 4, pp. 593–623, 1982.

[5] V. T. Chakaravarthy, V. Pandit, S. Roy, P. Awasthi, and M. Mohania, "Decision trees for entity identification: Approximation algorithms and hardness results," in *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2007, pp. 53–62.

[6] G. Blanchard and D. Geman, "Hierarchical testing designs for pattern recognition," *The Annals of Statistics*, vol. 33, no. 3, pp. 1155–1202, 2005.

[7] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4995–5004.

[8] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 39–48.

[9] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1–9.

[10] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[11] H. Qi, T. Wu, M.-W. Lee, and S.-C. Zhu, "A restricted visual turing test for deep scene and event understanding," *arXiv preprint arXiv:1512.01715*, 2015.

[12] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5014–5022.

[13] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 11–20.

[14] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," in *Proc. of CVPR*, 2017.

[15] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in VQA matter: Elevating the role of image understanding in visual question answering," in *CVPR*, vol. 1, 2017, p. 9.

[16] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678.

[17] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1261–1270.

[18] A. Newell and J. Deng, "Pixels to graphs by associative embedding," in *Advances in Neural Information Processing Systems*, 2017, pp. 2168–2177.

[19] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[20] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5831–5840.

[21] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende, "Generating natural questions about an image," *arXiv preprint arXiv:1603.06059*, 2016.

[22] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun, "ivqa: Inverse visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8611–8619.

[23] Y. Li, N. Duan, B. Zhou, X. Chu, W. Ouyang, X. Wang, and M. Zhou, "Visual question generation as dual task of visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6116–6124.

[24] A. Rothe, B. M. Lake, and T. Gureckis, "Question asking as program generation," in *Advances in Neural Information Processing Systems*, 2017, pp. 1046–1055.

[25] K. van Deemter, I. van der Sluis, and A. Gatt, "Building a semantically transparent corpus for the generation of referring expressions," in *Proceedings of the Fourth International Natural Language Generation Conference*. Association for Computational Linguistics, 2006, pp. 130–132.

[26] J. Viethen and R. Dale, "The use of spatial relations in referring expression generation," in *Proceedings of the Fifth International Natural Language Generation Conference*. Association for Computational Linguistics, 2008, pp. 59–67.

[27] M. Mitchell, K. van Deemter, and E. Reiter, "Natural reference to objects in a visual domain," in *Proceedings of the 6th international natural language generation conference*. Association for Computational Linguistics, 2010, pp. 95–104.

[28] M. Mitchell, K. Van Deemter, and E. Reiter, "Generating expressions that refer to visible objects," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), 2013.

[29] N. FitzGerald, Y. Artzi, and L. Zettlemoyer, "Learning distributions over logical forms for referring expression generation," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1914–1925.

[30] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 787–798.

[31] O. Frank and D. Strauss, "Markov graphs," *Journal of the american Statistical association*, vol. 81, no. 395, pp. 832–842, 1986.

[32] T. A. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock, "New specifications for exponential random graph models," *Sociological methodology*, vol. 36, no. 1, pp. 99–153, 2006.

[33] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, pp. 3618–3623, 2015.

[34] L. Younes, "Estimation and annealing for gibbsian fields," *Annales de l'IHP Probabilités et statistiques*, vol. 24, no. 2, pp. 269–294, 1988.

[35] ——, "On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates," *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 65, no. 3-4, pp. 177–228, 1999.

[36] A. L. Yuille, "The convergence of contrastive divergences," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 1593–1600.

[37] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines." in *AISTATS*, vol. 1, 2009, p. 3.

[38] W. A. Sands, B. K. Waters, and J. R. McBride, *Computerized adaptive testing: From inquiry to operation.* American Psychological Association, 1997.

[39] W. J. van der Linden and C. A. Glas, *Computerized adaptive testing: Theory and practice*. Springer, 2000.