# A top-down saliency model with goal relevance

**James Tanner**
Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

**Laurent Itti**
Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

**Most visual saliency models that integrate top-down factors process task and context information using machine learning techniques. Although these methods have been successful in improving prediction accuracy for human attention, they require significant training data and are unable to provide an understanding of what makes information relevant to a task such that it will attract gaze. This means that we still lack a general theory for the interaction between task and attention or eye movements. Recently, Tanner and Itti (2017) proposed the theory of goal relevance to explain what makes information relevant to goals. In this work, we record eye movements of 80 participants who each played one of four variants of a Mario video game and construct a combined saliency model using features from three sources: bottom-up, learned top-down, and goal relevance. We use this model to predict the eye behavior and find that the addition of goal relevance significantly improves the Normalized Scanpath Saliency score of the model from 4.35 to 5.82 ($p < 1 \times 10^{-100}$).**

## Introduction

Often, attention is discussed in terms of two facets: bottom-up and top-down. It is generally accepted that bottom-up represents attention drawn to visually distinct characteristics such as color and motion, whereas top-down attention is a tendency to look toward locations that provide more information based on the current context or goal of the viewer. Although bottom-up attention is well studied, our knowledge of top-down attention is not as advanced because of its more complex nature. Importantly, given a task, there is no general theory that allows researchers to quantitatively evaluate which pieces of information are more likely to attract the attention of humans performing that task. When driving, for example, humans are more likely to look toward intersections (Shinoda, Hayhoe, & Shrivastava, 2001). This is fairly intuitive, but there is no generalized method for explaining why this might happen.

Hayhoe and Ballard (2014) provided a review of recent advances in goal-directed attention research, in which they noted that attention has been examined in many visuomotor tasks, including walking, driving, sports, and making sandwiches (Hayhoe & Ballard, 2005; Land & Hayhoe, 2001; Land & Lee, 1994; Sprague & Ballard, 2003). The development of a general theory for top-down effects on eye movements has been avoided by the use of reinforcement learning techniques. For example, Navalpakkam and Itti (2005) developed one of the earlier computational models for predicting how tasks and goals might influence saccades. In their model, the task definition provides top-down biasing over which visual features would contribute more strongly to a saliency map for guiding attention. Given a task, their model first searches its long-term memory for relevant entities (where relevance is a conceptual distance over a concept ontology) and then biases attention toward the visual features of those entities. Items selected by attention are analyzed by an object recognition process, and any objects successfully identified in a scene are then added to the working memory of the model, allowing it to reason about both relevant entities and found entities when planning the next attention biasing and next eye movement. Although this model uses a simple classical similarity metric for the definition of relevance, it provides a computational framework in which relevance directly interacts with attention deployment.

Video games are increasingly being used in experiments because they allow for the presentation of complex visual tasks in a controlled environment. In one recent example, Kotseruba and Tsotsos (2017) presented an intricate cognitive architecture that combines mechanisms of visual attention, working memory, and executive control to play video games in real time and achieve scores comparable to human experts. Several studies have used learning from examples to implicitly capture task influences on eye

Citation: Tanner, J., & Itti, L. (2019). A top-down saliency model with goal relevance. *Journal of Vision*, *19*(1):11, 1–16, https://doi.org/10.1167/19.1.11.

https://doi.org/10.1167/19.1.11 Received December 18, 2017; published January 16, 2019 ISSN 1534-7362 

movements. In one such study, Peters and Itti (2007) recorded the human gaze while playing video games and learned a mapping from the visual input to screen locations. Borji, Sihite, and Itti (2014) also used video games to learn a Dynamic Bayesian Network for predicting future gaze. This model was able to outperform purely bottom-up saliency models in predicting a player's next eye movement. Their model essentially captures the current state of the game through analysis of past and present video frames, user actions, and eye movements. This allows it to learn associations between game state and relevant locations/objects, in which relevance is defined as the probability of being the location of the next fixation.

A shortcoming of this approach is that it is a black box, in that the learned parameters of the Dynamic Bayesian Network are not easily interpreted. Another is that the learning is task specific, so the model must be retrained for each new task. In addition, the process of learning does not single out top-down features. The learned model represents a complete approximation of attentional behavior including all factors. However, as argued by Pinto, van der Leij, Sligte, Lamme, and Scholte (2013), top-down information is likely independent of bottom-up. All of these points demonstrate that we still do not fully understand algorithmically how a given task specifically affects attention, despite the success of these studies. As a result, even the current state of the art in gaze prediction during tasks is unable to reliably mimic human gaze behavior (Borji et al., 2014). Thus, we do not have models that, given a task, can generally predict a priori which objects will attract more attention.

Additional recent approaches to top-down saliency models have aimed to supply category-specific saliency models (i.e., provide one saliency map for an image focusing on a cat and another map for the same image focusing on a bottle instead). This approach allows the appropriate map to be chosen based on the object of interest. In Kocak, Cizmeciler, Erdem, and Erdem (2014), object classes are represented using superpixels, which are learned in a conditional random field (CRF) framework. Another variant (Yang & Yang, 2017) uses sparse coding from images patches, which are also learned in a CRF. These methods provide excellent performance, but they are machine learning models and thus still leave us without any conceptual understanding of top-down attention. Also, these methods are essentially solving an object detection task and therefore will have difficulty on tasks in which the contextual information is more than a simple object of interest.

The problem with these models (Borji et al., 2014; Peters & Itti, 2007) is that they learn to predict eye movement behavior without having any understanding of the process from which that behavior emerges. Although their predictive accuracy is higher than models without learned top-down features, the top-down features represent a very shallow understanding, and there is much room for improvement. The learned models also present a requirement to be retrained on each new task and cannot predict a priori which objects will attract more attention. Developing an understanding of the relationship between goal relevance and eye movements can hopefully address these issues.

Recently, Tanner and Itti (2017) proposed a simple mathematical definition of goal relevance, which suggests that an observation is relevant to a task to the degree to which it affects the solution space of the task. As described in Tanner and Itti (2017), goal relevance is defined for a data observation $D$ with respect to an agent's probability distribution $P(S)$ over the set $S$ of possible ways it could achieve its goals as a distance measure, $d(\cdot, \cdot)$, between the prior distribution of beliefs $P(S)$ and the posterior distribution $P(S|D)$ after observation of data $D$:

$$\mathcal{R}(D, S) = d(P(S|D), P(S)) \quad (1)$$

In their article, the Kullback-Leibler divergence is used as the distance metric for $d(\cdot, \cdot)$, along with a discretized form of the function, giving

$$\begin{aligned}\mathcal{R}(D, S) &= KL(P(S|D), P(S)) \\ &= \sum_S P(S|D) \log \frac{P(S|D)}{P(S)} \quad (2)\end{aligned}$$

Tanner and Itti (2017) demonstrated that goal relevance was correlated with human responses to a simple visual task. This equation still requires that the task be defined in terms of its solution space, but it provides a quantitative value for task relevance. Our contribution is to establish that goal relevance is also correlated with human eye movements while performing a task and to use it in a top-down saliency model. Goal relevance as defined here is closely related to the previously proposed mathematical definition of surprise (Itti & Baldi, 2006). In this experiment, we make the assumption that participants will spend more time fixating objects that are more relevant to the task. Objects of interest might first be noticed in peripheral vision and fixated to gather more details or the dynamic nature of the world might require multiple fixations toward highly relevant objects in order to monitor them for changes.

# Methods

## Experimental methods

Human participants were recruited and their eye movements (saccades) recorded while performing a

task. Participants sat 106 cm away from a 42-in., 1,920 × 1,080 pixel LCD monitor screen and images subtended approximately 45.5° × 31° visual angle. Similar to the experiment presented in Peters and Itti (2007), the task was a video game. However, in this experiment, all participants played a single game, Mario World, and were be told that their goal was to reach the end of each level as quickly as possible while taking as little damage as possible (specifically, they were told to maximize their score, which will be explained later). Participants were first briefed and then given a short questionnaire asking about their level of experience with Mario World and with video games in general. Then, after calibration with the eye-tracking equipment, they had the opportunity to practice playing the game until they were ready. Finally, they played through eight levels in a randomized order. Their eye movements were recorded during this period in addition to during the practice period. In total, 84 participants were recruited (42 males and 42 females).

Mario World was chosen because of a competition held several years ago, called MarioAI ("2012 Mario AI Championship," n.d.), which released an open-source Java implementation of the game. This allows us to modify Mario World for our own purposes, to create a customized levels to best test our hypotheses and also adjust some of the game mechanics. For readers familiar with Mario World, the mechanics we have changed are as follows:

1. Mario has unlimited life. When he takes damage, the participant receives feedback but Mario is not killed.
2. There is no time limit for completing a level.
3. None of the levels contain any pits into which Mario could fall and die.
4. All coins and power-ups have been removed from the game.
5. Mario is unable to run and can only walk.
6. When a Koopa enemy is destroyed, no shell is left behind.
7. At the end of each level, an itemized score is displayed based on the time taken and how many times Mario took damage.

Some of these changes were to ensure consistency across participants. Changes 1 to 3 guaranteed that all participants reached the end of the customized levels and therefore saw all of the content in them. Changes 4 to 6 removed some of the more complex aspects of the game, so that there would be less of a difference in performance between participants who were new to the game and those who had prior experience. In particular, Change 5 prevented very experienced participants from completing the levels in a drastically shorter time than other participants, so there would be less variation in the overall session duration.

Change 7 exists to make explicit what the participants' goals were as they played. This allows us to confidently design a predictive model using the same goals as the human participants. The score displayed to participants was based on two factors: the amount of time taken to complete the level and the number of times that Mario took damage during the level. The total score was calculated using the following equation:

$$\text{Score} = 100{,}000 - (t \times 100) - (d \times 3{,}000) \quad (3)$$

where $t$ is the amount of time taken, in seconds, and $d$ is the number of times Mario received damage. This equation means that a single instance of taking damage costs the same amount of points as spending an additional 30 seconds. Our custom levels are designed such that there is no scenario in which this occurs, so it is never worth taking damage to reach the end of the level more quickly. The optimal strategy is to prioritize avoiding damage and otherwise proceed as quickly as possible. This was explained to the participants, and the score display after each level showed the calculations to continually reiterate this point. The intention of this scoring metric was to ensure that the participants focused on avoiding the enemies and obstacles, rather than simply running as fast as possible without regard for the environment. To this end, in observing the participants, we felt we were successful in keeping them highly engaged in safely navigating the levels.

Unbeknownst to the participants, they were each randomly assigned to one of four different versions of our game before they arrived, such that each group contained 21 participants. The different versions are as follows:

- Normal: Our modified Mario World game as described above.
- Small: Mario is half as tall as in the normal version. This allows him to fit through some gaps not accessible to normal Mario.
- High Jump: Mario can jump about twice as high as normal Mario.
- Invulnerable: Mario does not take damage when coming into contact with enemies and instead simply passes through them.

Participants were made aware of the special properties of their version of the game but were not told about the other versions. The assigned version was used for the entirety of the session, both during practice and data collection.

Each version of the game is designed to have a significant impact on the space of possible actions that Mario can take, and the levels are also designed to focus on these differences. For example, in many locations throughout the levels, there are enemies high above the ground such that they cannot interact with normal Mario (Figure 1A). In this case, the theory of

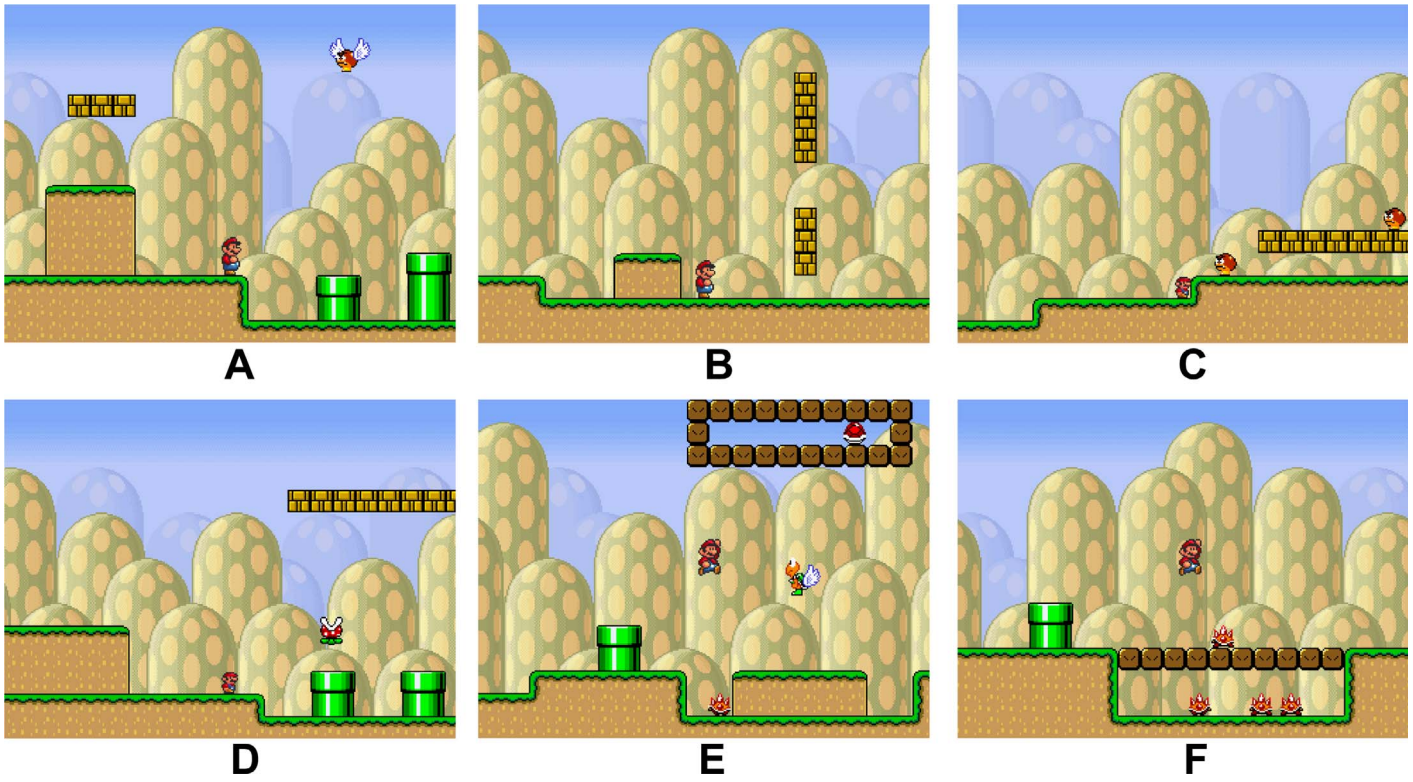


Figure 1. More goal relevance test object examples. (A) The enemy Goomba is too high to threaten normal Mario. However, high-jumping Mario must be careful to jump over the pipe while not colliding with the enemy. (B) Normal Mario must go through the middle opening in the wall. However, small Mario can also choose to walk under the bottom, and high-jumping Mario can leap over the wall. (C) Small Mario can avoid some enemies by walking under the walkway, but the other Marios cannot fit. (D) High-jumping Mario can jump on top of the overhanging walkway to avoid an enemy. (E) The shell inside the blocks at the top is highly salient from a bottom-up perspective because it bounces back and forth quickly and makes noise. However, no version of Mario can interact with it. (F) The three Spiky enemies at the bottom are also uninteractable.

goal relevance labels this enemy as irrelevant to the participant's goal. However, high-jumping Mario must be careful not to jump while passing underneath this enemy to avoid taking damage. For participants assigned to the High Jump version, this enemy was considered relevant. Examples such as this are littered throughout the levels, where goal relevance will assign different values of relevance to certain objects depending on which version of Mario World is being played (Figure 1). In this way, we can investigate a causal effect of changing the solution space on eye movement behavior.

## Analytical methods

With the data collected, two separate analyses were performed: (1) determining if changes in goal relevance values caused by changes in the solution space are correlated with the amount of time participants spend looking at those objects and (2) using the goal relevance model as an additional feature in a saliency model to see if the predictive accuracy can be improved. These analyses will be referred to as the Solution Space Experiment and the Saliency Model Experiment, respectively.

The data analysis for both of these experiments requires the computation of goal relevance for game objects in each recorded video frame. To use Equation 2, a goal and a solution space for Mario World must first be defined. In Mario World, the goal is to reach the end of the level as quickly as possible and while taking damage the least number of times. However, because of the side-scrolling nature of Mario World, the end of each level is not visible on screen until Mario has almost reached it. The participant is also completely unaware of any enemies and obstacles until Mario has progressed far enough for them to enter the visible scene. Therefore, we instead define our goal as reaching the right side of the visible scene as quickly as possible and while taking damage the least number of times. This better matches the goal toward which participants will most likely plan, because they do not have any information with which to plan further.

The Mario game operates in frames, at a rate of 24 frames per second. A solution for reaching the goal is a list of discrete actions to be taken in each frame. As

there are only three different buttons used to control Mario (move left, move right, and jump), there are six distinct actions:

1. Do nothing
2. Move left
3. Move right
4. Jump
5. Jump left
6. Jump right

To enumerate the possible solutions, we perform a breadth-first search, simulating the game state for each action at each time step, and recording any path through the breadth-first graph that ends with a node representing Mario reaching the right side of the screen. Of course, there are an infinite number of such solutions because of the option to do nothing for any amount of time, and aside from that, this graph would expand exponentially and become intractable very quickly. To avoid both of these issues, we first compute the optimal solution using the A* search algorithm (Hart, Nilsson, & Raphael, 1968), using the following heuristic for evaluating the fitness $h$ of each node in the graph:

$$h(x, y, t, d) = 100,000 - ((t + e(x)) \times 100) - (d \times 3,000) \quad (4)$$

where $x$ and $y$ are the coordinates of Mario in the level, $t$ is the amount of time currently elapsed, in seconds, and $d$ is the number of times Mario has received damage. $e(x)$ is a function for estimating the amount of time remaining before Mario reaches the level end and is simply calculated as the remaining distance to the goal divided by Mario's maximum speed. Note that this equation is the same as Equation 3, except that the time component is broken down into current time and estimated remaining time, which is available to the model. The players had no direct information about the estimated remaining time. However, all of the levels are the same length, so players could have noticed this and gained an approximate sense. The model computes the optimal solution to Equation 4 using A*, which is used to limit the breadth-first search by evaluating each node using the same fitness function. Any node whose fitness is lower than the optimal solution's fitness by more than a threshold $\delta_{\text{fitness}}$ is pruned, reducing the graph to a manageable size. Even with this restriction, the size of the search tree becomes extremely large, and the search must be performed several times on every frame. Because of this, it became necessary to set this threshold to 0 to satisfy computational constraints. To ensure that this did not negatively affect our results, we compare the results of our saliency models using a higher threshold, but using data from a small subset of our participants, in the Effect of $\delta_{\text{fitness}}$ on NSS Scores subsection.

After obtaining a list of possible solutions, a model space must be chosen that can represent them as a distribution. Similar to the procedure in Tanner and Itti (2017), we discretize the visible screen into a two-dimensional grid and project the solutions onto it, in which each grid cell is assigned a value equal to the number of solutions passing through that cell. The resulting distribution can be compared (using Equation 2) to another distribution that does not include the object in question to produce the goal relevance of the object. This full process is shown in Figure 2.

The last decision to make with regard to computing goal relevance is what should be considered an object for which we will evaluate goal relevance. It is clear that we should compute goal relevance for each enemy, but the method for evaluating the relevance of blocks and terrain is not obvious. For blocks, we decided to consider each connected component (groups of adjacent block tiles) as a single object. The same strategy cannot be used for terrain because all of the terrain in an entire level would be considered as a single connected component, and we want to capture the different sections of the terrain. Instead, we evaluate individual grid units of terrain at corners: locations where a unit of terrain is bordered by nonterrain on the top and either the left or the right. Each terrain corner is the same size as a standard block unit. Overall, this lets our model consider many possible places of interest.

We additionally compute the goal relevance of Mario himself. However, a game state without Mario included would have no solutions, which would lead to a null distribution over which Kullback-Lieblier divergence (KL) is undefined. This same problem occurs when there are no paths to a goal. Tanner and Itti (2017) suggested that this situation be handled by adding a fixed, uniform low-probability density over the solution space. We could alternatively use a uniform probability over the reachable states rather than the whole space, but the intuition of this distribution in terms of reaching the goal is unclear. Therefore, the goal relevance of Mario is computed by comparing a flat distribution over the whole space with the prior. Figure 3 visualizes the objects on which goal relevance is computed, along with the result.

### Solution space experiment

As noted earlier and shown in Figure 1, throughout the game levels are many objects designed to have significantly different values for goal relevance depending on which version of Mario World the participant has been assigned, which will be referred to as *test objects*. This first experiment uses these to see if changing the solution space causes a predictable change in human gaze behavior.

To show that changing the solution space causes differences in eye movements, we need to measure the goal relevance and the amount of time spent looking at

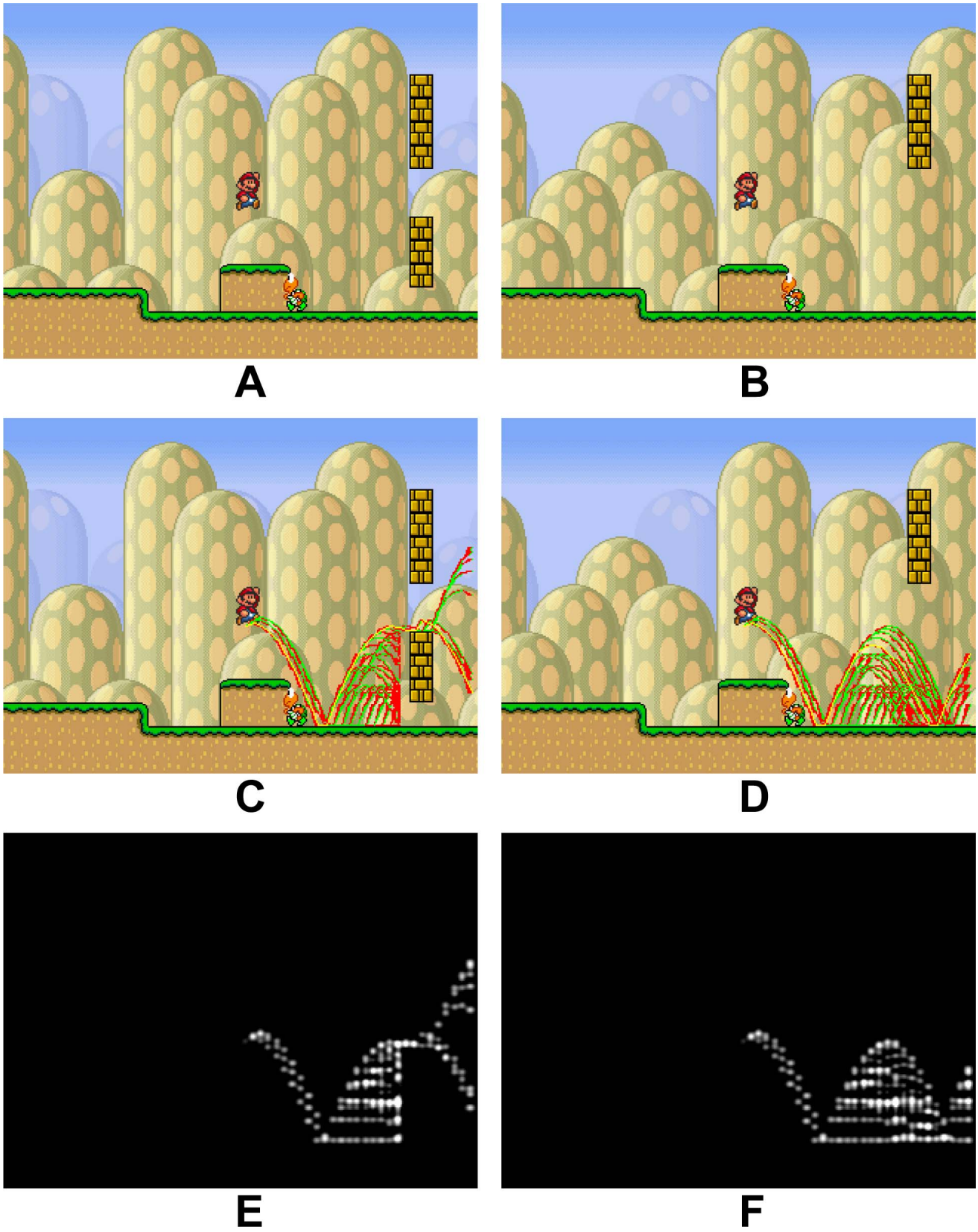


Figure 2. Visualization of goal relevance computation. In this case, we are computing the goal relevance of the lower set of blocks. (A, B) First, a posterior state is created in which the object in question is removed. (C, D) Then, the solution set for each state is found using the guided breadth-first search described above. Paths are colored from green to red based on their fitness compared with the →

←

optimal path found by A*, shown in yellow. Mario is too large to fit through the gap at the bottom, so he must jump through the middle. (E, F) The solution sets are converted into probability distributions via grid discretization. These two images have been edited with increased gamma and blurring to improve visualization. Lastly, the goal relevance value is determined by applying Equation 2 to the two distributions.

many objects and see if changes in these averages across game versions are correlated between them. For example, we expect that enemies hovering high in the air (as in Figure 1A) will have much higher goal relevance values for participants playing the High Jump version, and we hypothesize that this will lead to those same participants spending significantly more time looking at those enemies than participants in the other groups.

To measure this, for each participant and each test object, we compute two values, referred to as the attention ratio (AR) and the goal relevance ratio (GRR). The AR represents how much time that participant spent attending to that object, and the GRR represents the average goal relevance of the object.

$$\mathrm{AR} = \frac{\text{Number of frames attending to the object}}{\text{Number of frames object is visible}} \quad (5)$$

$$\mathrm{GRR} = \frac{\text{Sum of goal relevance of the object for each frame it is visible}}{\text{Number of frames object is visible}} \quad (6)$$

The equations are designed this way to account for the fact that participants progress through the game at different speeds. Even though each participant sees every object, the objects are not visible for the same amount of time for each participant. Participants who are slower would have more frames in which to view the objects, so we must average based on the number of frames in which the objects are visible. Also note that the goal relevance values are different in each frame, depending on the game state, so they must also be averaged. The subject is considered to be attending to the object if his or her gaze location is within a threshold distance $\delta_{\text{test}}$ of the object's center, to account for the fact that subjects often attend to locations adjacent to intended objects and for minor eye-tracking miscalibrations that arise from small head movements. We set $\delta_{\text{test}} = 4°$ visual angle, which is approximately the height of Mario. These AR and GRR values are then averaged across participants, separately for participants in each of the four versions of the game. This will provide us with four AR averages and four GRR averages for each object. Finally, we analyze these values within specific subsets of objects. Specifically, we investigate the following object subsets:

1. Flying enemies (Figure 1A)
2. Custom wall lowers: The lower portion of the specially designed wall sections (Figure 1B)

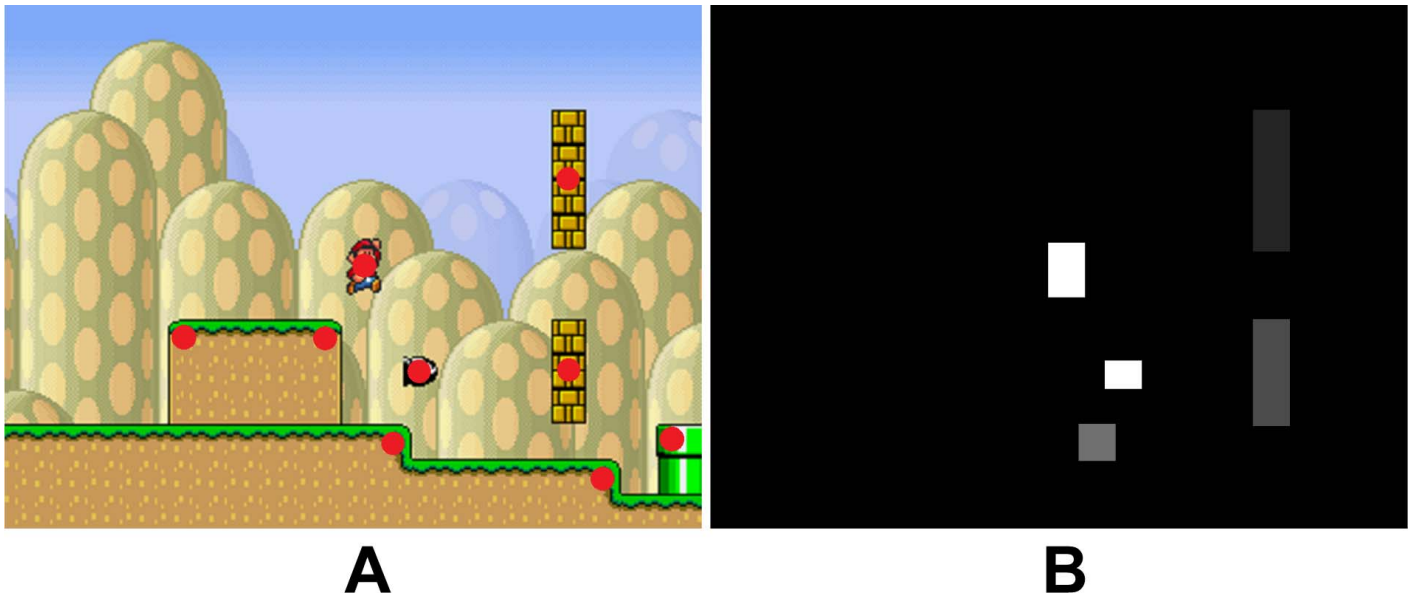


Figure 3. Objects on which goal relevance is computed. (A) Each red dot indicates an object to which we assign a goal relevance value. Goal relevance is computed on enemies, terrain units at corners, blocks, and on Mario himself. (B) The results of the goal relevance computations, visualized as a raw saliency mask. The goal relevance of an object is distributed over its visual area in the mask. This is the same mask that will be used in our saliency model. In this case, the two most relevant objects are Mario and the Bullet Bill enemy. Note that some objects do not appear in the mask because of low or 0 values for goal relevance.

3. Custom wall uppers: The upper portion of the specially designed wall sections (Figure 1B)
4. Tunnels: Long horizontal blocks under which only small Mario can fit (Figure 1C)
5. Raised ledges: Blocks onto which only high-jumping Mario can jump (Figure 1D)
6. Uninteractable enemies: Enemies hidden behind blocks such that no version can interact with them (Figure 1E,F). Flying enemies are also considered uninteractable for game versions other than High Jump
7. Interactable enemies: All enemies not considered uninteractable

### *Saliency model experiment*

The next experiment aims to show that goal relevance can be used to improve predictive accuracy in saliency models. Specifically, our aim is not to show state-of-the-art results but to demonstrate that goal relevance presents a new theory of top-down attention that can significantly contribute to a combined model. Our saliency model is a combination of three separate models: Itti and Koch's classic bottom-up model (Itti, Koch, & Niebur, 1998), the learned top-down model presented in Peters and Itti (2007; their full model also uses Itti et al. [1998], but for clarity, we include the two parts separately), and our top-down goal relevance model. The bottom-up model compares distributions of low-level features (luminance, color opponency, and orientation) at multiple scales and identifies conspicuous locations based on image statistics. The learned top-down model first computes a reduced feature vector for each image representing the "gist," using the same multiple-scale low-level features. It then learns a mapping between this vector and likely gaze locations. This model is trained separately for each individual participant and is trained and tested using a leave-one-out scheme by game level so that each of the participant's levels is evaluated by a model trained on all of the other levels played by the same participant.

To construct the goal relevance saliency mask for each frame of the recorded video, the goal relevance values are first computed as described previously. Then, for each object, a bounding box in the image is computed, and its goal relevance is distributed evenly across the values in the corresponding box of the mask. This means that a small object will have a larger pixel-wise value than a larger object with the same total goal relevance. Distributing the goal relevance in this way accounts for the fact that some objects are larger on the screen and ensures that the total presence of an object in the mask is proportional to its goal relevance value. An example mask is shown in Figure 3. This goal relevance mask will be provided to the combined model, in addition to the other masks produced by the bottom-up and learned top-down models. Each mask is individually blurred to maximize performance, using a standard Gaussian filter with a sigma of 40 pixels and a kernel size of 161 pixels.

To combine these three masks into a single model, we take a simple linear combination of the masks and their second-order terms. That is, given masks BU, TD, and GR, the combined mask can be computed with the following equation:

$$\begin{aligned} C = {} & x_0 \times \mathrm{BU} + x_1 \times \mathrm{TD} + x_2 \times \mathrm{GR} \\ & + x_3 \times \mathrm{BU} \times \mathrm{TD} + x_4 \times \mathrm{BU} \times \mathrm{GR} \\ & + x_5 \times \mathrm{TD} \times \mathrm{GR} + x_6 \times \mathrm{BU} \times \mathrm{TD} \times \mathrm{GR} \end{aligned} \quad (7)$$

where $x_0 - x_6$ are coefficients selected using the simplex search method (Lagarias, Reeds, Wright, & Wright, 1998), an iterative algorithm for minimizing nonlinear functions. To avoid overfitting, we randomly set aside one participant from each group (two men, two women) and use the results from these four participants in the simplex search. The remaining analysis proceeds with these participants excluded, leaving 20 (10 men, 10 women) in each group.

A number of metrics have been proposed for evaluating saliency models, from normalized scanpath saliency (NSS) and several variants of area-under-the-curve (AUC) to the correlation coeffiecient (CC), KL, and, most recently, information gain (IG; Kümmerer, Wallis, & Bethge, 2015) metrics. For two reasons, we have decided to primarily report our results using the NSS score (we also computed our results using the AUC metric, which can be found in the AUC Results subsection of the Appendix). First, typical saliency models operate on data sets of static images, in which each participant views each image for a period of time. This provides a distribution over the image for each participant of attended locations and allows for a correspondence between participants. In this experiment, we used dynamic video (controlled by real-time user input), which provides us with only a single point for each image and prevents any correspondence between frames. This makes the creation of a human competitor model as a benchmark impossible. Also, several of the saliency metrics are designed to compare between two distributions and thus are not as well suited for comparing a model prediction to a single point. The CC, KL, and IG scores fall into this category. Second, there has been much discussion in the saliency community about the growing number of metrics and the inconsistencies between them. In their review, Bylinskii, Judd, Oliva, Torralba, and Durand (2016) recommended KL or IG when evaluating probabilistic saliency models and the NSS and CC metrics when using saliency models for capturing viewing behavior. Because our experiment falls into the latter category, we chose NSS.

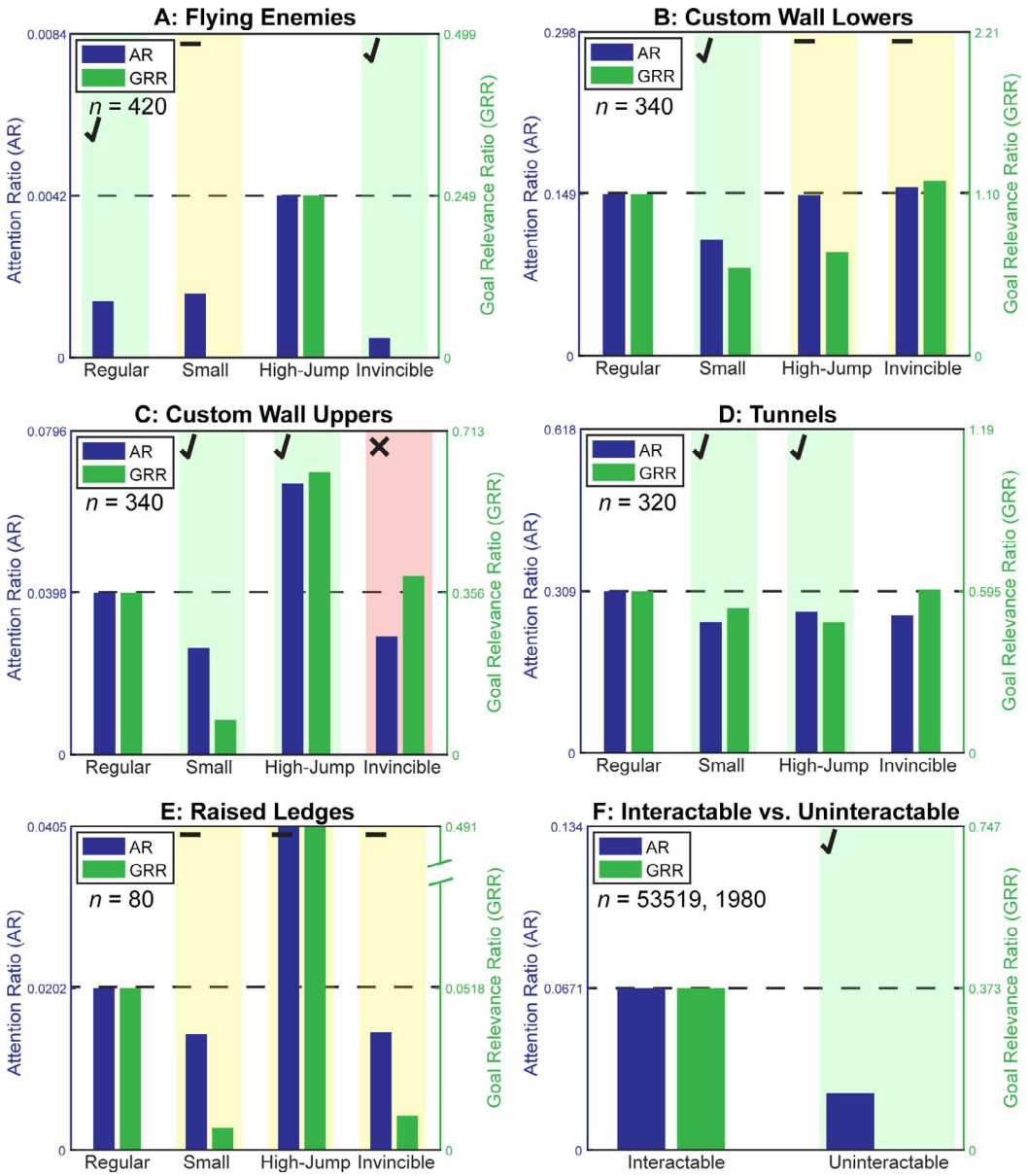


Figure 4. Attention ratio (AR) and goal relevance ratio (GRR) averages for specific subsets of objects. These are the same object subsets shown in Figure 1A-F. For easier comparison, a baseline game version is selected (usually regular Mario), and the axes are scaled such that the AR and GRR values for the baseline are centered. The horizontal dotted line represents the baseline. In cases in which the GRR is significantly different from baseline, columns are highlighted in green (✓) if the AR is significantly different in the same direction, red (×) if the AR is significant in the opposite direction, and yellow (line) otherwise. (E) Note the axis break that is necessary to maintain the normalized view. (F) Instead of comparing game versions, this graph compares interactive and noninteractive enemies, averaged across versions. The differences sometimes appear small because, despite our intent to measure top-down effects, bottom-up effects on eye behavior are still present, which means that in some cases, the relative effect of the game version on AR can be small.

# Results

## Solution space experiment

The hypothesis of this experiment is that by manipulating the game rules and thereby controlling the solution space and the goal relevance values, a similar change in human attention would be produced. To evaluate this hypothesis, we analyzed how the GRR and AR values change in one version of the game compared with another for specific object subsets. Specifically, for each object category, a baseline version was selected, and we measured whether significant differences in GRR values were matched by significant differences in AR values in the same direction. The results of this analyses are shown in Figure 4. In cases in which the GRR values

| | Versus | GRR | AR |
|---|---|---|---|
| Flying | R | $t(838) = 12.18$, $p = 1.58 \times 10^{-31}$ | $t(838) = 2.00$, $p = 4.54 \times 10^{-2}$ |
| | S | $t(838) = 12.18$, $p = 1.58 \times 10^{-31}$ | $t(838) = 1.52$, $p = 0.128$ |
| | I | $t(838) = 12.18$, $p = 1.58 \times 10^{-31}$ | $t(838) = 2.83$, $p = 4.80 \times 10^{-3}$ |
| | S | $t(678) = 22.62$, $p = 7.56 \times 10^{-85}$ | $t(678) = 5.13$, $p = 3.86 \times 10^{-7}$ |
| Wall lowers | HJ | $t(678) = 13.71$, $p = 6.15 \times 10^{-38}$ | $t(678) = 0.14$, $p = 0.89$ |
| | I | $t(678) = -3.45$, $p = 6.06 \times 10^{-4}$ | $t(678) = -0.68$, $p = 0.50$ |
| | S | $t(678) = 31.18$, $p < 1 \times 10^{-100}$ | $t(678) = 2.54$, $p = 0.01$ |
| Wall uppers | HJ | $t(678) = -18.93$, $p = 1.82 \times 10^{-64}$ | $t(678) = -3.88$, $p = 1.14 \times 10^{-4}$ |
| | I | $t(678) = -3.80$, $p = 1.55 \times 10^{-4}$ | $t(678) = 1.88$, $p = 0.06$ |
| | S | $t(638) = 6.69$, $p = 4.92 \times 10^{-11}$ | $t(638) = 4.35$, $p = 1.58 \times 10^{-5}$ |
| Tunnels | HJ | $t(638) = 11.13$, $p = 2.04 \times 10^{-26}$ | $t(638) = 2.93$, $p = 3.50 \times 10^{-3}$ |
| | I | $t(638) = -0.62$, $p = 0.54$ | $t(638) = 3.37$, $p = 8.00 \times 10^{-4}$ |
| | S | $t(158) = 8.96$, $p = 8.56 \times 10^{-16}$ | $t(158) = 0.64$, $p = 0.52$ |
| Ledges | HJ | $t(158) = -31.34$, $p = 1.06 \times 10^{-69}$ | $t(158) = -1.87$, $p = 0.06$ |
| | I | $t(158) = 7.75$, $p = 1.07 \times 10^{-12}$ | $t(158) = 0.63$, $p = 0.53$ |
| Interactable | Uninteractable | $t(55497) = 24.75$, $p < 1 \times 10^{-100}$ | $t(55497) = 27.46$, $p < 1 \times 10^{-100}$ |

Table 1. Statistical results using a standard *t* test, corresponding to the data in Figure 4. *Notes*: For each object category, a game version selected as the baseline is compared against the other three versions, in both GRR and AR. R = regular; S = small; HJ = high jump; I = invincible.

in a version are significantly different from the baseline GRR, we show the following:

1. A green shade, if the AR values are significantly different from the baseline AR in the same direction as the GRR difference (up or down)
2. A yellow shade, if the AR values are not significantly different from the baseline
3. A red shade, if the AR values are significantly different but in the opposite direction as GRR

We used the Regular Mario version as the baseline in most cases. Note that in Figure 4A, the High Jump version is used as the baseline because the GRR values for all other versions are 0. All of the statistical results for this figure are shown in Table 1. Of 16 comparisons (three in each of A–E, one in F) we show eight green shades, six yellow shades, one red shade, and one with no shade (Figure 4D Invincible, in which the GRR was not significantly different from baseline).

Figure 4A shows the values for all flying enemies across the four game versions. As high-jumping Mario is the only version capable of interacting with these enemies, the GRR values are trivially 0 for all other versions, and so the GRR values for all the other versions were statistically different from the baseline. The AR values also show intuitive results, following the same trend by being lower for the nonbaseline versions. However, the AR values for the small version did not quite meet the 5% confidence level for significance, so a yellow shade is shown. This is caused by a very high standard deviation in AR values, which is also present in the other object subsets.

In Figure 4B, we have a similar story, although a bit less intuitive. For the lower section of the custom walls, the GRR value is significantly lower for small Mario because small Mario can fit under the wall, so removing the wall has less of an impact on the solutions. We can see that this is matched with the significantly lower AR value. The GRR is also lower for high-jumping Mario because he can jump over the entire wall and ignore the bottom section, but the AR in that case is only a bit lower, and it is not significant.

The custom wall upper sections (Figure 4C) were similar to A, with the highest GRR values for the high-jumping version matched by the highest AR values. The higher goal relevance comes from these objects being higher in the air and mostly reachable only by high-jumping Mario. The invincible version for this subset was the only case in which GRR and AR values were significantly different in opposite directions.

Figure 4D shows the results of what we refer to as tunnels, which are long sequences of blocks that leave just enough space underneath for small Mario to pass through. The initial prediction for these objects was that small Mario would have the lowest GRR because he is affected the least, just as with the lower wall sections. The small Mario cases indeed had a lower GRR, but it turns out that high-jumping Mario has the lowest GRR. This is due to high-jumping Mario being able to spend more time in the air, in which blocks near the ground have less of an effect. No shade is shown for the invincible version because the GRR values were not significantly different from baseline.

Figure 4E shows three yellow shades, even though the AR values matched the changes in GRR values, particularly for the high jump version, in which we should intuitively see the biggest change. Unfortunately, the AR differences did not reach significance, in

part because of the smaller sample size for this object subset because the raised ledges were the least common of the specialized objects in our levels. We suspect that at least the high jump version, if not all three versions, would become significant with enough data. However, for now, we can only say that there is a trend here.

Finally, Figure 4F is different in that it compares two object subsets rather than a single subset across the different versions. However, the results are clear; noninteractive enemies, which trivially have a goal relevance of 0, are also looked at significantly less frequently than interactive enemies, which have a significantly higher goal relevance.

In addition, we computed the correlation between AR and GRR values for all objects. Across all objects presented in the experiment, there was a weak but significant correlation between AR and GRR values, $r(162{,}218) = 0.1217$, $p < 1 \times 10^{-100}$. This correlation is impressive because the AR computation, unlike the GRR, captures all influences on human gaze, not just the top-down influences. This means that many of the saccades may be guided by other factors that goal relevance is not designed to capture. It is well known that task plays a role in attention (Hayhoe & Ballard, 2014), but it is clearly not the only factor, so it is expected that there be a significant portion of the AR values that are a priori uncorrelated with the GRR. Because of this, the GRR values are likely much more strongly correlated with the human notion of goal relevance than indicated by the numbers alone.

## Saliency model experiment

As described earlier, we also used goal relevance to create a saliency mask, which was included in a combined model along with masks from bottom-up and learned top-down models. Figure 5 visualizes several example frames and the masks produced by each of the three components individually, along with the combined model mask. Overall, goal relevance generally provides more intuitive masks, focusing on enemies and objects with which Mario will soon interact. However, human participants also spend significant time looking at other parts of the scene, such as less relevant objects or the background, and these cases are better captured by the learned top-down model regardless of intent.

The metric results of the models are shown in Figure 6. Here we show the model results only for the NSS metric, but results for the AUC metric can also be found in the AUC Results subsection of the Appendix. To compare the contribution of the three models, we show results for each model individually as well as the three combinations of the two models together. For example, a combined model including only BU and TD is one in which $x_2$ and $x_4 - x_6$ are set to 0 and $x_0$, $x_1$, and $x_4$ are found using the simplex search method as above. These combinations allow us to see the effects of adding or removing individual components from the combined model.

Although goal relevance is not intended to be used as a stand-alone saliency model, we can still gain insights from comparing the individual models. First, it is clear that TD performed the best of the three models, based on its individual score and on the scores of combinations in which it is included. Goal relevance performs well, and every combination in which it is included has a higher score than the corresponding combination without it. For example, the TdGr combination does much better than TD on its own.

The BU model struggled with this task, with the individual BU model barely scoring above chance. This may be caused by the biases present in this particular task. For example, there was an extremely strong bias for participants to attend to the right side of the screen, because their goal is to move Mario to the right, and therefore, the right side contains the information needed for planning a path. The TD model can handle this bias because it learns where the participants tended to look, and the GR model discovers this on its own because the paths with higher fitness all go toward the right. The BU model has no way to account for this, so it gives just as much weight to salient things on the left side as on the right side, significantly decreasing its score. However, we can see that the information provided by the BU model is still useful because combining BU with the other models yields improvements. This happens because of the second-order terms, in which the BU mask is multiplied by other masks. In these terms, the right-side bias is already contained in the additional masks and BU simply filters for the most salient regions remaining.

The higher scores from the GR model suggest that this task had a strong effect on eye behavior. Specifically, it indicates that participants reliably gazed toward objects that directly affected how the task could be solved. The goal relevance calculations were able to capture this, including the right-side bias, purely based on a computational definition of the task itself. It is interesting that goal relevance was a reliable predictor of eye behavior even in the presence of very strong distractors. For example, some of the noninteractive enemies (such as the shells bouncing back and forth) are extremely distracting because of the speed at which they move and the sounds they make. In terms of bottom-up saliency for the human, they are often the most salient objects on the screen, and it is quite reasonable to expect participants to look at them frequently. Goal relevance, by contrast, gives them a value of 0 because they do not affect any solutions. Although the participants did occasionally look at

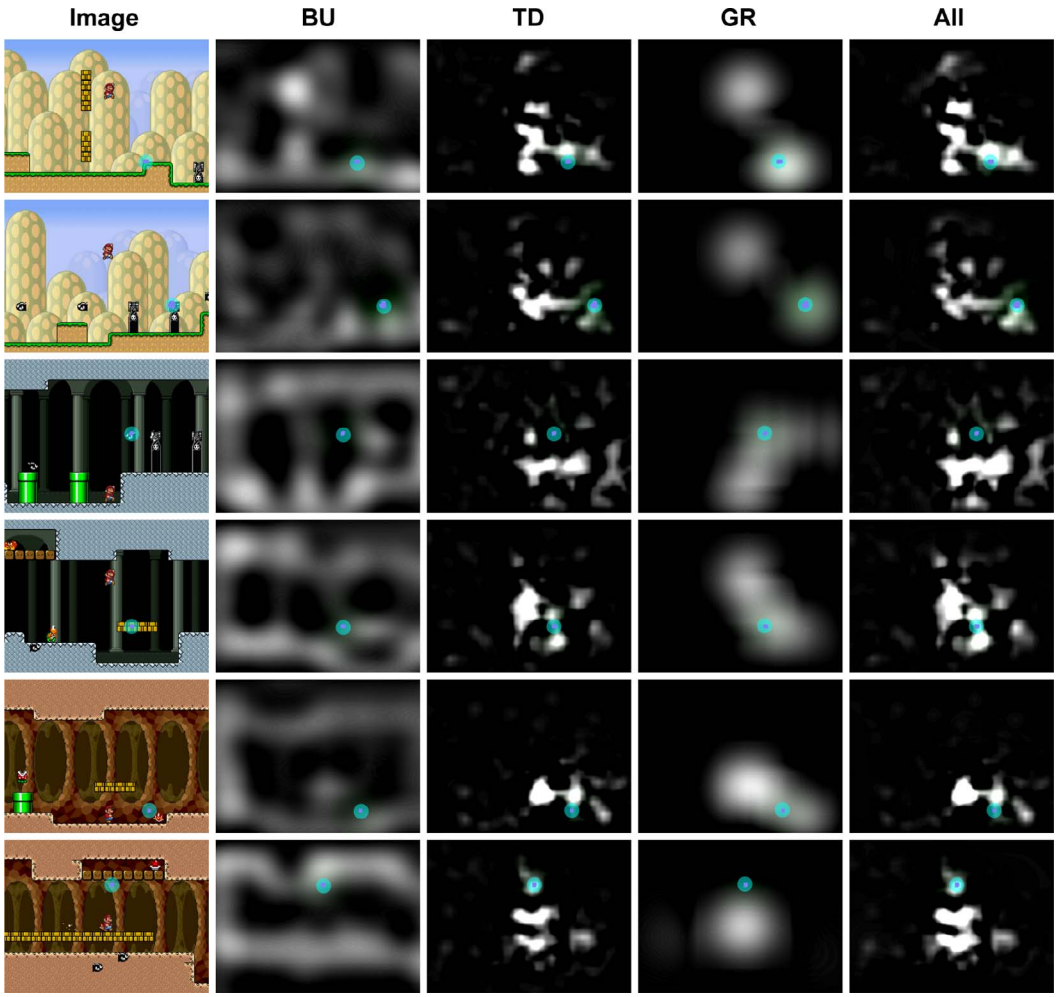


Figure 5. Example images along with the model prediction masks. We show the mask produced by each individual model (including blurring) along with the final combined model. In each image, the cyan dot marks the participant's gaze location. Note that here we show the final masks, which include blurring, particularly the BU and GR masks. Also, mask images have been edited with increased gamma to improve visibility. In most of these examples, goal relevance does very well. For each row, starting from the top: (1) Gaze is directed to a ledge as Mario falls. (2) Gaze is directed to a cannon which fires missiles. The left-most cannon is not considered relevant because Mario is in the ascent of his jump, so most solutions land beyond it. (3) Gaze is directed toward an enemy. (4) Gaze is directed toward a platform onto which Mario will soon land. (6) The last example shows a situation in which the TD model performs much better. Gaze is being directed toward the background above Mario. Goal relevance is incapable of predicting this location because there are no objects there, but the learned model places a low probability there.

these enemies, Figure 4F, in addition to the NSS scores, shows that the difference in goal relevance had a very strong effect on the time spent looking at them.

The TD model performed the best because it is able to directly learn where participants tend to look. In this case, it learned the right-side bias and also that participants frequently looked toward a particular spot slightly below the center of the screen, which can be seen in the TD masks in Figure 5. The game screen scrolls to the side as Mario progresses through each level, such that Mario is always centered horizontally, and this spot is based on the average height of Mario. The model also learns other regions of the screen where each participant tends to look, although on average it does not give them as much weight as this below-center location.

Finally, introducing an additional component caused a significant improvement in the NSS score in all cases. This is a pleasing result that suggests that none of the models overlaps another in terms of the information it provides. In particular, goal relevance performed well and significantly contributed to all of the models in which it was included. There was a significant effect of adding the GR mask to the BU mask to form the BuGr model, $F(279,648) = -313.1510$, $p < 1 \times 10^{-100}$. Similarly, adding GR to the TD mask was also significant, $F(279,648) = -91.2175$, $p < 1 \times 10^{-100}$, as well as adding GR to the BuTd

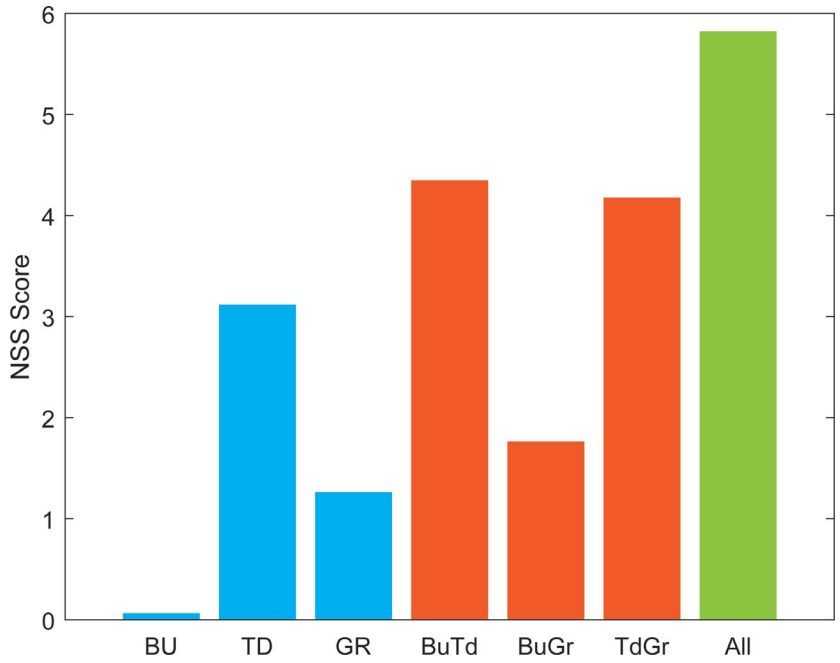


Figure 6. NSS scores for all combinations of the three model components: bottom-up, top-down learned, and goal relevance. In all cases, a combination that includes goal relevance performs significantly better than the corresponding combination without goal relevance.

model, $F(279,648) = -89.9835$, $p < 1 \times 10^{-100}$. These results also hold for each game version individually, indicating that there was not much difference between versions in terms of the NSS scores.

To confirm that our results do not only apply to a single environment representation, we repeated the above analysis where gaps between blocks or the ground were also interpreted as objects. Goal relevance for these gaps was computed by simulating an environment with the gap filled in by blocks. A common example of where this occurs can be seen in Figure 1B, between the upper and lower custom wall block segments and between the lower segment and the ground. This modified analysis caused the performance to drop slightly for each model that included goal relevance. However, this reduction was less than 0.6% of the NSS score in the GR model and even less in the combined models that included GR. None of the statistical results were affected.

### Game experience questionnaire

At the beginning of each session, the participant was asked to rate the amount of prior experience they had playing Mario video games from 1 to 4 (1 being barely any or no experience, 4 being significant prior experience). We found a significantly positive correlation between these responses and the total game score received by the participants, $r(78) = 0.31$, $p = 5.1 \times 10^{-3}$. However, filtering the model results based on this question did not reveal any significant differences in model performance based on player experience. That is, experience level affected the score but did not significantly affect the agreement between our model predictions and the players' eye movements.

## Discussion

In Tanner and Itti (2017), goal relevance was designed to measure the degree to which information pertains to a task, but that does not necessarily imply that it would be correlated with eye behavior. However, all of the results of the present study together suggest that goal relevance is indeed correlated with eye movements. Almost all object subsets tested in Figure 4 produced results correlating goal relevance and visual attention. Half of the comparisons were green, indicating a positive correlation, while only a single comparison indicated the opposite. Figure 4 also shows that changing the task itself causes a change in goal relevance that is matched by changes in eye behavior. This is a promising result that might indicate goal relevance can generalize to other tasks. Finally, the significant correlation between GRR and AR across all objects directly showed that goal relevance is correlated with eye movements.

Furthermore, Figure 6 demonstrates the effectiveness of goal relevance as a predictor of eye behavior in a saliency model. The fact that goal relevance improved

the model for each combination in which it was included suggests that the predictive information provided by goal relevance is distinct from both the bottom-up and top-down learned features. This supports our belief that goal relevance provides a new potential stream of information that can be incorporated into any saliency model involving a specific task. The best models can be formed by combining multiple distinct sources of information, including learned features and conceptually bound features such as those of the BU and GR models.

Even though the learned top-down model had the highest individual performance based on NSS scores, the fact that the goal relevance represents distinct information means that it will generally improve the results of any model in which it is included. Also, goal relevance performs well despite the significant advantages of machine learning techniques and comes without their disadvantages. Goal relevance requires no training data at all, whereas most learned top-down models (including this one) learn from eye behavior of the same participant. The TD model is not capable of making a priori predictions about new participants or new tasks, but the goal relevance model handles this no differently than examples it has seen before. Also, even though we refer to the learned model as a top-down model, its learning process does not discriminate information. While learning top-down information, it also captures bottom-up and any other category of information that is present in the data, so it is really an all-around learned model.

For this particular task, the authors suspect that the primary advantage of the learned model comes from its ability to adapt to the visuals, rather than the participant. It is able to learn more precisely which locations on the screen are important and make small adjustments based on visual cues. For example, when there are many hills on screen so that the ground is elevated, the model often shifts its predictions slightly upward. Even though the model was trained separately for each participant, there did not seem to be any similarly clear differences in the model predictions between participants. This does not rule out more subtle adjustments, which could still have a significant effect on the prediction accuracy.

Despite the correlations between goal relevance and eye movements in this study, there are some caveats to this approach. One such issue is that goal relevance still requires specification by the researcher to define the task solution space and to designate which pieces of information should be evaluated for relevance. Every task is different, and there is no way to escape the fact that tailoring computation to a task requires some additional information. Goal relevance can be thought of as a conceptual framework that provides a simple way to think about tasks and information. This is similar to Itti and Baldi's (2006) concept of surprise, which also defined a novel way of thinking about bottom-up attention but left open the question of which features to use. Of course, this similarity is not an accident, as the two concepts share much of the same formulation.

A minor issue is that during the training process, participants assigned the invincible condition very quickly realize that the game can be played optimally by simply holding the "move right" button and occasionally pressing "jump." It is possible that this scenario simplifies the game such that subjects become less engaged and begin issuing saccades to less important areas of the screen to entertain themselves. However, this is purely speculative.

The choices of how to model the solution space and which objects should be evaluated for goal relevance undoubtedly had a large impact on the goal relevance values. There is not always an obvious answer to what should constitute a single piece of information for the purposes of this calculation. When deciding how to handle the blocks, for example, we could instead have evaluated all of the pieces (individual tiles) of each block individually. Also, we could have evaluated the relevance of empty spaces by comparing with game states containing blocks inserted at the same locations. This decision, along with the representation of the solution space, can be thought of as encapsulating the subjective nature of the task. None of these choices is any more inherently correct than another, and ultimately, we chose to only include existing objects whose goal relevance would be evaluated by deleting them and exclude empty spaces. We chose to group blocks into connected components because it was more intuitive to interpret the game environment in that way. This is why we also tested an alternative representation that included the gaps between blocks. Changing the representation did not affect our results, which suggests that goal relevance has some degree of robustness to the precise representation chosen. It also demonstrates that two humans with different representations can be equally capable of reaching a goal.

Another important choice was the manner in which goal relevance values were converted into a saliency mask to be used in the model. As goal relevance is still a very new concept, the most straightforward method was chosen, simply using object bounding boxes filled with intensities proportional to the goal relevance values. This implies that a very basic logic scheme is being used, namely, that gaze is directed toward the most relevant pieces of information. This includes no inhibition of return and does not account for the many subproblems that compose complex tasks. For example, even though the overall goal was to reach the end of a level, a participant might briefly become focused specifically on jumping onto a single ledge. This could

lead them to focus on the positioning of the ledge and on Mario, ignoring other enemies and obstacles that are still relevant but are beyond the ledge. Future work might improve upon goal relevance by developing a hierarchy of goals and subgoals to account for cases such as this one.

Our results do not imply that the details of how goal relevance is computed match the implementation in the human brain. Indeed, one might be doubtful that human participants simulated thousands of game states at each moment to decide where to direct their gaze. This is certainly another direction for future research, but we believe it has been established that goal relevance is related to the same result as the human computations, even if the brain arrives via a different manner. It is possible that humans develop heuristics (Gigerenzer, 2008), in which case we would argue that goal relevance is the target that the heuristics attempt to approximate.

## Conclusions

Overall, this study demonstrates that goal relevance provides a complete theory for the top-down behavioral effects of tasks on human gaze, from deciding which information is relevant to the task to how that information affects gaze behavior. In addition, it can be introduced into saliency models, even those that already include learned top-down features, to improve their performance. This is an important step toward improving our understanding and models of top-down gaze in humans.

*Keywords: top-down, relevance, attention, modeling, goals*

## Acknowledgments

This work was supported by the National Science Foundation (CCF-1317433 and CNS-1545089), the Office of Naval Research (N00014-13-1-0563), and Intel Corporation. The authors affirm that the views expressed herein are solely their own and do not represent the views of the United States government or any agency thereof.

Commercial relationships: none.
Corresponding author: James Tanner.
Email: jetanner@usc.edu.
Address: Viterbi School of Engineering, University of Southern California Los Angeles, CA, USA.

## References

2012 Mario AI Championship. (n.d.). Retrieved from http://www.marioai.org/. Accessed March 22, 2017.

Borji, A., Sihite, D. N., & Itti, L. (2014). What/where to look next? modeling top-down visual attention in complex interactive environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*, 523–538.

Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., & Durand, F. (2016). What do different evaluation metrics tell us about saliency models? *arXiv preprint arXiv:1604.03605*.

Gigerenzer, G. (2008). Why heuristics work. *Perspectives on Psychological Science*, *3*, 20–29.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, *4*, 100–107.

Hayhoe, M., & Ballard, D. (2005). Eye movements in natural behavior. *Trends in Cognitive Sciences*, *9*, 188–194.

Hayhoe, M., & Ballard, D. (2014). Modeling task control of eye movements. *Current Biology*, *24*, R622–R628.

Itti, L., & Baldi, P. (2006). Bayesian surprise attracts human attention. *Advances in Neural Information Processing Systems*, *18*, 547.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 1254–1259.

Kocak, A., Cizmeciler, K., Erdem, A., & Erdem, E. (2014). Top down saliency estimation via superpixel-based discriminative dictionaries. In *Proceedings British Machine Vision Conference 2014*. BMVA Press. http://dx.doi.org/10.5244/C.28.73

Kotseruba, I., & Tsotsos, J. K. (2017). Star-rt: Visual attention for real-time video game playing. *arXiv preprint arXiv:1711.09464*.

Kümmerer, M., Wallis, T. S., & Bethge, M. (2015). Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, *112*, 16054–16059.

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, *9*, 112–147.

Land, M. F., & Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision Research*, *41*, 3559–3565.

Land, M. F., & Lee, D. N. (1994, June 30). Where we look when we steer. *Nature*, *369*, 742.

Navalpakkam, V., & Itti, L. (2005). Modeling the influence of task on attention. *Vision Research*, *45*, 205–231.

Peters, R. J., & Itti, L. (2007). Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *IEEE Computer Vision and Pattern Recognition*. (pp. 1–8). Washington, DC: IEEE.

Pinto, Y., van der Leij, A. R., Sligte, I. G., Lamme, V. A., & Scholte, H. S. (2013). Bottom-up and top-down attention are independent. *Journal of Vision*, *13*(3):16, 1–14, https://doi.org/10.1167/13.3.16. [PubMed] [Article]

Shinoda, H., Hayhoe, M. M., & Shrivastava, A. (2001). What controls attention in natural environments? *Vision Research*, *41*, 3535–3545.

Sprague, N., & Ballard, D. H. (2003). Eye movements for reward maximization. In *Advances in neural information processing systems* (pp. 1467–1474).

Tanner, J., & Itti, L. (2017). Goal relevance as a quantitative model of human task relevance. *Psychological Review*, *124*, 168.

Yang, J., & Yang, M.-H. (2017). Top-down visual saliency via joint CRF and dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 576–588.

# Appendix

## Effect of $\delta_{\text{fitness}}$ on NSS scores

In the Analytical Methods subsection, we described our method for dealing with an infinite solution space, which was to prune the search tree by removing any node with a fitness value more than $\delta_{\text{fitness}}$ below the fitness of the best path found using an A* search. However, computational restraints required that we set this threshold to 0, meaning that we pruned branches as soon as they became worse than the best solution. To confirm that this did not affect our results, we reran the analysis using higher values but only on the four validation participants used for the simplex search (Lagarias et al., 1998). This is shown in Figure 7. Although the NSS scores did slightly worse with the higher values, the difference is small enough that the significance of the results is not affected.

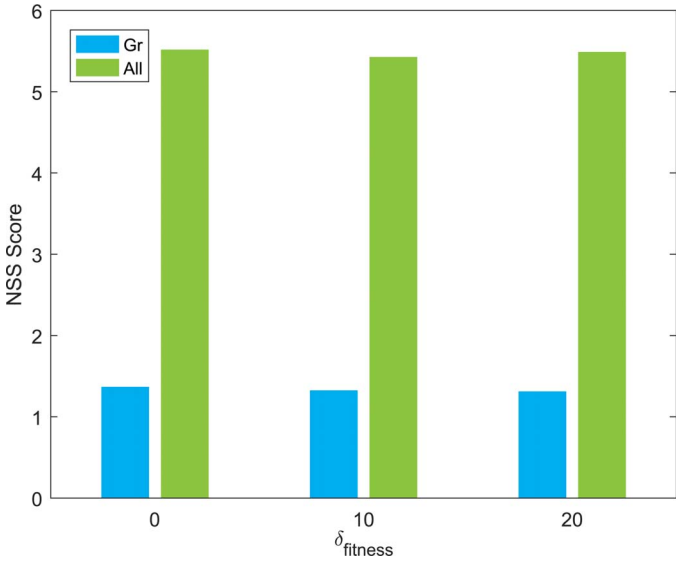


Figure 7. NSS scores for the GR and full combined models for different values of $\delta_{\text{fitness}}$. The difference in the scores is very small, so using a low value does not affect our results.

## AUC results

In Table 2, we show the model scores for the same models shown in Figure 6, except that we evaluate the models using the AUC metric instead. Just as before, there was a significant effect of adding the GR mask to the BU mask to form the BuGr model, $F(279{,}648) = -269.6313$, $p < 1 \times 10^{-100}$. Adding GR to the TD mask was also significant, $F(279{,}648) = -6.8460$, $p = 7.60 \times 10^{-12}$, as well as adding GR to the BuTd model, $F(279{,}648) = -6.7976$, $p = 1.07 \times 10^{-11}$. Although the differences between the model scores are not as large, the significance of the results is unaffected.

| Component | AUC score |
|---|---|
| BU | 0.5282 |
| TD | 0.9088 |
| GR | 0.7973 |
| BuTd | 0.9088 |
| BuGr | 0.7982 |
| TdGr | 0.9137 |
| All | 0.9137 |

Table 2. AUC scores for all combinations of the three model components: bottom-up, top-down learned, and goal relevance. *Notes*: In all cases, a combination that includes goal relevance performs significantly better than the corresponding combination without goal relevance.