Supplementary Material for: TIAM - A Metric for Evaluating Alignment in Text-to-Image Generation

Paul Grimal Hervé Le Borgne Olivier Ferret Julien Tourille

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{paul.grimal, herve.le-borgne, olivier.ferret, julien.tourille}@cea.fr

1. Proof of Proposition 1 and its Generalization

We give the proof of proposition 1, which is easy to establish, then we derive a formula giving the number of templates in a more general realistic case (proposition 3). For this, we introduce an intermediate step (proposition 2) and most importantly a set of notations that facilitates the derivation of the final analytic formula.

Let us consider a collection of object sets $\{\mathcal{O}_i\}_{i=1}^{N}$ and a collection of attribute sets $\{\mathcal{A}_i\}_{i=1}^{N}$ that are used to define a template with N objects, each being qualified by an attribute. Hence, at position i (without considering the context), the prompt results from the inference of the template and contains a named object $o_i \in \mathcal{O}_i$ that is qualified by a (color) adjective $a_i \in \mathcal{A}_i$.

Proposition 1: if $|mO| \geq N$, $|\mathcal{A}| \geq N$ and $\forall i \in [\![1, N]\!], \mathcal{A}_i = \mathcal{A}, \mathcal{O}_i = \mathcal{O}$ and $\forall (i,j) \in [\![1, N]\!]^2$ s.t $i < j$, we force $a_i \neq a_j$ and $o_i \neq o_j$, thus the number of unique prompts generated by the template is $\frac{|\mathcal{O}|!|\mathcal{A}|!}{(|\mathcal{O}|-N)!(|\mathcal{A}|-N)!}$

Proof. Without attribute, each prompt contains N objects that should be different; thus the number of unique possible prompts is the number of (arrangements) N-permutations of $|\mathcal{O}|$ thus $\frac{|\mathcal{O}|!}{(|\mathcal{O}|-N)!}$. Similarly, if one considers the attributes only that must be different at each position i, we have N-permutations of $|\mathcal{A}|$. Finally, since both are independent, the final number of unique prompts generated by the template is the product of both. □

We can consider a generalization where the sets are different at each position, that is we remove the conditions $a_i \neq a_j$ and $o_i \neq o_j$ from proposition 1. For example, such a template could be defined to have a vehicle at the first position, a fruit at the second position, and an animal at the third one. The intersection of the sets at each position may also be non-empty, for example, the vehicle may be $\{blue, red, green, black, yellow, white\}$, the fruit $\{red, green, yellow\}$, and the animal $\{green, black, yellow, white\}$. Similarly, the attributes can be repeated in the prompt *e.g.* if one wants "a yellow car and a red apple and a red elephant".

Proposition 2: if $\forall i \in [\![1, N]\!]$, $|\mathcal{O}_i| \geq N$, $|\mathcal{A}_i| \geq N$ thus the number of unique prompts generated by the template is $\prod_{i=1}^{N} |\mathcal{A}_i|.|\mathcal{O}_i|$. Hence, if $\forall i \in [\![1, N]\!], \mathcal{A}_i = \mathcal{A}, \mathcal{O}_i = \mathcal{O}$, thus the number of unique prompts generated by the template is $(|\mathcal{O}|.|\mathcal{A}|)^N$

Proof. At any position i any all the attributes $a_i \in \mathcal{A}_i$ can be associated with the object $o_i \in \mathcal{O}_i$ thus it gives $|\mathcal{A}_i|.|\mathcal{O}_i|$ possibilities. Since repetitions are allowed, we have a structure of arborescence (rooted tree), thus the total number of prompts is the product at any position, thus $\prod_{i=1}^{N} |\mathcal{A}_i|.|\mathcal{O}_i|$. □

However, if the sets of objects are not exclusive we consider that we should not allow any repetition, since it would be strange to require "a car and an apple and a car". Such repetition can nevertheless be allowed if the object appears several times with different attributes. In other words, if $\forall i \in [\![1, N]\!]$, $|\mathcal{O}_i| \geq N$, $|\mathcal{A}_i| \geq N$ and $\forall (i,j) \in [\![1, N]\!]^2$ s.t $i < j$ we force $o_i \neq o_j$ if $a_i = a_j$.

In that case, things depend on the overlap between the combination of attributes and objects at each position. In other words, if the attributes are colors, it depends on how many times the same colored object can appear at different positions in the prompt. Let us consider the sets $\mathcal{U}_i = \mathcal{A}_i \times \mathcal{O}_i$ of *objects with attributes* at position i in the prompt. Since we have $o_i \neq o_j$ if $a_i = a_j$, each $\mathcal{U}_i$ can be split into two disjoint subsets. The subset $\mathcal{U}_i^0$ contains the objects with attributes that can appear at position i only, and its complement in $\mathcal{U}_i$ contains the objects with attributes that can appear at another position $j \neq i$ in the prompt.

The number of prompts that the template can generate depends on the overlap between these complements of the $\mathcal{U}_i^0$. For instance, the "blue car" can appear at positions 1 and 3, while the "green apple" can appear at positions

$\mathcal{U}_1$	$\mathcal{U}_2$	$\mathcal{U}_3$
blue truck	green banana	green tiger
red truck	purple banana	blue tiger
blue bike	green cherry	green bear
red bike	purple cherry	blue bear
red car	purple apple	yellow apple
blue car		blue car
	green apple	green apple

Table 1. Example of *objects with attributes*, above dashed line: that can appear at a specific location and are thus in $\mathcal{U}_i^0$; below dashed line: that can appear at several locations in the prompt. One notes that in that case $\mathcal{O}_1 = \{'truck','bike','car'\}$ and $\mathcal{A}_1 = \{'blue','red'\}$

.

configuration			$\mathcal{P}$
pos. 1	pos. 2	pos. 3	
blue car	green apple	$\mathcal{U}_3^0$	$\{3\}$
blue car	$\mathcal{U}_2^0$	green apple	$\{2\}$
blue car	$\mathcal{U}_2^0$	$\mathcal{U}_3^0$	$\{2,3\}$
$\mathcal{U}_1^0$	green apple	blue car	$\{1\}$
$\mathcal{U}_1^0$	green apple	$\mathcal{U}_3^0$	$\{1,3\}$
$\mathcal{U}_1^0$	$\mathcal{U}_2^0$	blue car	$\{1,2\}$
$\mathcal{U}_1^0$	$\mathcal{U}_2^0$	green apple	$\{1,2\}$

Table 2. Illustration of the set $\mathcal{P}$ (fourth col.) for the example of Tab. 1. The notation $\mathcal{U}_i^0$ means that any colored object of $\mathcal{U}_i^0$ can be used at that position

2 and 3 as illustrated in Tab. 1. The objects that can appear at several positions define some configurations that are uniquely defined by the set of positions where they appear in the prompt. Symmetrically, these configurations are also defined by the sets of positions where only one object appears at a single position (thus from $\mathcal{U}_i^0$). We note $\mathcal{P}$ the set of these last positions (See Tab. 2 for an illustration of these sets). Note that the same element can appear several times in $\mathcal{P}$, actually when several objects with attributes can appear at the same position (*e.g.* both "blue car" and "green apple" can appear at position 3 in our example). Formally, $\mathcal{P}$ is a set that contains the sets of indexes of the non-empty positions of all the N-uples that verify $(\mathcal{U}_i \backslash \mathcal{U}_i^0) \cap (\mathcal{U}_j \backslash \mathcal{U}_j^0) \neq \emptyset$. Using this formalization of the problem, we have:

Proposition 3: Let $\forall i \in \llbracket 1, N \rrbracket$, $|\mathcal{O}_i| \geq N$, $|\mathcal{A}_i| \geq N$. If $\forall (i,j) \in \llbracket 1, N \rrbracket^2$ s.t $i < j$ and $a_i = a_j$, we force $o_i \neq o_j$, thus the number of unique prompts generated by the template is:

$$\prod_{i=1}^{N} |\mathcal{U}_i^0| + \sum_{P \in \mathcal{P}} \prod_{i \in P} |\mathcal{U}_i^0| \qquad (1)$$

Proof. Since the elements of the $\mathcal{U}_i^0$ are different and appear at a unique position, they generate $\prod_{i=1}^{N} |\mathcal{U}_i^0|$ templates.

For any configuration in $\mathcal{P}$, an object with attributes (from one of the $\mathcal{U}_i \backslash \mathcal{U}_i^0$) appears at most once since if $a_i = a_j$ thus $o_i \neq o_j$.

For each configuration $P \in \mathcal{P}$ (such as the lines of Tab. 2) the number of templates generated is the product of all the $|\mathcal{U}_i^0|$ for that configuration, thus $\prod_{i \in P} |\mathcal{U}_i^0|$ for each configuration of position P. If we sum for all the possible P in $\mathcal{P}$, it results in the number of templates generated with objects with attributes that can appear at several positions.

The sum of both terms gives the value in Equation 1 □

Hence the exact number of prompts generated depends on the possible overlaps between objects with attributes, both the number of elements and the place they can appear or not. By introducing more notations on these elements and their position, one could derive a formula but it would be tedious, without obvious interest in practice. Indeed, $\mathcal{P}$ can be built easily through a tree structure, by considering iteratively all the sets $(\mathcal{U}_i \backslash \mathcal{U}_i^0) \cup \{\emptyset\}$, and allowing only the nodes that did not previously appear and the branch corresponding to a non-empty set in $\mathcal{P}$ (such as in Tab. 2).

2. Prompt Templates

We detail the templates and the COCO labels used in the study.

2.1. Without attribute

The 24 COCO labels used for the study of Sections 4.1 and 4.2 are the following: *bicycle, car, motorcycle, truck, fire hydrant, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, banana, apple, broccoli, carrot, chair, couch, oven, refrigerator*. The prompt template is "a photo of $det(o_1)$ o_1 and $det(o_2)$ o_2".

For the semantic studies, end of Section 4.3, we use the vehicles, animals, and food labels *i.e. bicycle, car, motorcycle, airplane, bus, train, truck, boat, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake*. The template used is "a photo of $det(o_1)$ o_1 and $det(o_2)$ o_2".

To evaluate the capacity of models to represent multiple objects, we use the following template :

- 1 object: "a photo of $det(o_1)$ o_1"
- 2 objects: "a photo of $det(o_1)$ o_1 and $det(o_2)$ o_2"
- 3 objects: "a photo of $det(o_1)$ o_1" next to $det(o_2)$ o_2 and $det(o_3)$ o_3"
- 4 objects: "a photo of $det(o_1)$ o_1" next to $det(o_3)$ o_2 with $det(o_3)$ o_3 and $det(o_4)$ o_4"

Model	w comma	w/o comma
IF	0.12	0.21 $(+0.09)$
SD 1.4	0.01	0.02 $(+0.01)$
SD 1.4 A&E	0.13	0.17 $(+0.04)$
SD 2	0.13	0.15 $(+0.02)$
SD 2 A&E	0.17	0.21 $(+0.04)$
unCLIP	0.13	0.10 (-0.03)

Table 3. Comparison of the TIAM for two templates with 3 entities, separated by commas or related with words.

We do not use commas because it tends to reduce the score. We report in Tab. 3 the comparison between the prompt "*a photo of* $det(o_1)$ o_1 *next to* $det(o_2)$ o_2 *and* $det(o_3)$ o_3" and the prompt with a comma "*a photo of* $det(o_1)$ o_1, $det(o_2)$ o_2 *and* $det(o_3)$ o_3".

2.2. With Attribute

To evaluate the attribute binding in Section 4.4, we used the following templates:

- one object : "a photo of $det(a_1)$ a_1 o_1"
- two objects : "a photo of $det(a_1)$ a_1 o_1 and $det(a_2)$ a_2 o_2 "

As reported in Sections 3.2 and 4.4 of the main paper, we have $\mathcal{O} = \{$*car, refrigerator, giraffe, elephant, zebra*$\}$ and $\mathcal{A} = \{red, green, blue, purple, pink, yellow\}$.

3. Reference Colors and Other Possible Attributes (Size, Texture)

We plot in Fig. 2 the interpolation of the best examples in Lab on the CIE 1931 Chromaticity Diagram. The exact chroma values were extracted from Fig.1 in [9], available at this link. We do not use our reference colors *orange* and *brown* because they are too close to *red* and *black* respectively in the CIELab space (Fig. 1).

To consider an attribute in TIAM, two crucial points need to be considered:

- being able to extract the attribute from the image, with a sufficient level of reliability
- being able to name the attributes with unambiguous words in the prompt, ideally in several languages

The attribute *colors* have the advantage of quite easily meeting these two conditions, as explained in Section 3.2 of the main paper, in particular, thanks to the works of Berlin and Kay [1]. In other cases, however, this may prove trickier.

To extract the attribute *size*, one can rely on the bounding box of the object detector. However, to determine whether an object is *large*, *medium*, or *small* for example, it may also require to estimate its depth in the image (that can be done from monocular images to a certain extent [17]), possibly an estimation of the intrinsic parameter matrix of the camera (while the image is generated) as well as the knowledge of the typical dimensions of the considered object or living being. Each of these estimations is a potential source of approximation that challenges the reliability of the final judgment.

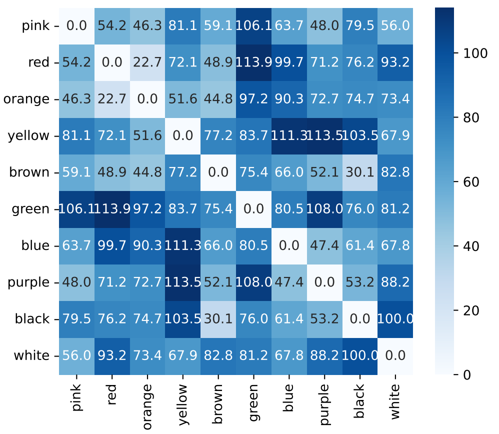


Figure 1. L2 norm distance between our reference colors.

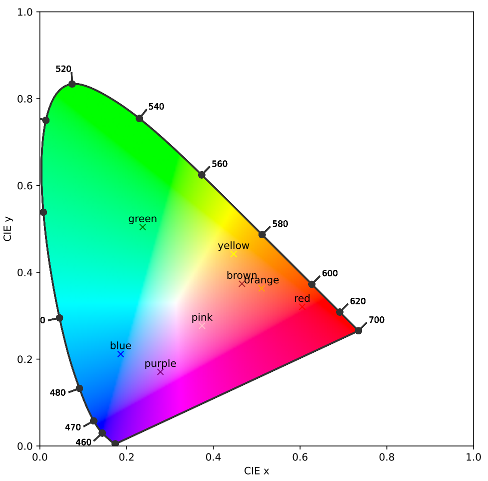


Figure 2. CIE 1931 Chromaticity Diagram with the best example for each color. When multiple best examples for one color, we compute one best example by averaging the value in the CIELAB space.

Naming the *size* may also be diverse. In the example above, *large* may be replaced by *big* in English and *small* by *tiny* in some cases. One could imagine relying on lists of synonyms or setting a threshold on the similarity to the textual embedding (*e.g*. BERT) of an arbitrary predefined list of possible sizes (*e.g*. large, medium, small), but there is no guarantee to get a list as unambiguous as in the case of colors. To our knowledge, there is no equivalent of the study of Berlin and Kay in that case. Moreover, references to sizes tend to include intensifiers more frequently than references to colors, like in expressions such as very small or fairly large, which adds an element of diversity. Finally, the way sizes are expressed can depend on forms of collocations. For instance, we can refer to a "tall man" but not to a "tall balloon". While this example can be probably explained by the form of the object, the easiest way to deal with such problem would be to collect co-occurrences from a corpus for (size adjective, noun) pairs and to use these co-occurrences as a filter after the generation of a prompt.

Extracting texture is a long-term and well-known task in computer vision [8] and many methods have been proposed to address it [7]. The question of naming the texture with unambiguous names may seem delicate. In the famous Brodatz dataset for example (available here), one can note that several of them are named *Wood shingle roof* (here and href), *Brick wall* (here, here, or here) or *Sand* (here, here and here) among others. Bhushan *et al.* (1997) nevertheless identified a list of 98 representative words used to describe texture in English [2] that was further reduced to 47 in the Describable Textures Dataset [3]. However, such a number remains quite large (much more than the 11 colors of Berlin and Kay), such that the correspondence in other language than English is hazardous, not to mention the fact that some of them may seem ambiguous to several human users (our human study in Section 11 includes users less than 10 years old with a limited vocabulary, as well as a majority of persons that are not computer vision scientists). Using Multidimensional Scaling (MDS) on these 98 words, Bhushan *et al.* (1997) nevertheless identified 11 clusters that could be used, although naming these clusters is still problematic in practice. Finally, the easier approach may be to use three axes, identified with the MDS as well, namely:

- repetitive versus nonrepetitive textures
- the nature of orientation: linearly oriented textures → multiple or no orientation → circularly oriented textures.
- complexity or simplicity of the surface

Hence, considering other attributes than *colors* in TIAM is likely feasible, but integrating them neatly into the metric may require some work.

4. Occurrence of Objects on Images

We show in our experiments (Section 4.3) that the initial objects in the template tend to appear more frequently than objects inserted subsequently and reinforce the observation that the concept that is expressed earlier in the prompt has more chances to appear in the final image. We present the result for two objects in Fig. 3 and three objects in Fig. 4.

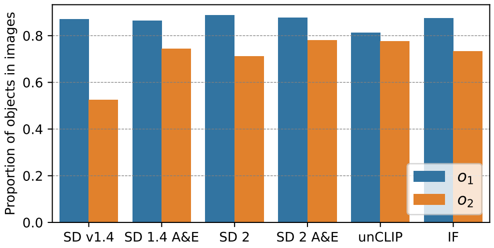


Figure 3. The proportion of occurrences of each object, based on its position in the prompt. The template of the prompt includes two objects.

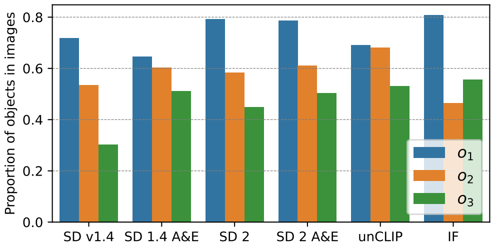


Figure 4. The proportion of occurrences of each object, based on its position in the prompt. The template of the prompt includes three objects.

5. Determining the Minimum Number of Images to Generate

In Fig. 5 we report the TIAM score as a function of the number of generated images per prompt. The score stabilizes from 16 images. We chose to compute with 32 images to ensure robustness.

6. Detection/Segmentation Details

We use the largest YOLOv8 for segmentation[1]. During segmentation inference, we set up the object confidence threshold for detection to 0.25 and the intersection over union IoU threshold for NMS to 0.8. We compute the *score* with different confidence thresholds and observe that the score decreases linearly as the threshold values increase (Fig. 6).

[1] https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8x-seg.pt

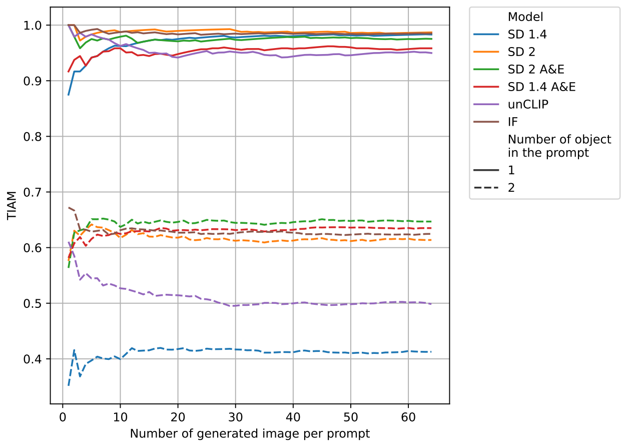


Figure 5. TIAM as the function of the number of generated images per prompt.

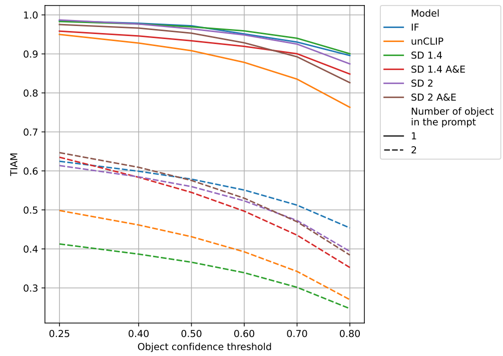


Figure 6. TIAM as a function of the YOLO object confidence threshold.

7. Text-to-Image Models Setup

All generations of images were done on Nvidia A100 SXM4 80 Go using `float16`. We list the main parameters for the different models used. If not mentioned, we use the default parameters from the library diffusers [19] (version 0.16.1).

SD 1.4 [2] **and SD 2** [3] The models produce images of size 512×512.

- Guidance scale: 7.5
- Scheduler : DPMSolverMultistepScheduler [4] [14]
- 50 inference steps

unCLIP [5] The model produces images of size 256×256.

- 25 inference steps for the prior, 25 inference steps for the decoder, and 7 steps of Super-resolution
- Prior guidance scale: 4, decoder guidance scale: 8
- Scheduler is the UnCLIPScheduler, a modified DDPM scheduler designed for this model.

IF It exists a different configuration of the Deepfloyd IF. We use for the first stage the L version[6] with 100 inference steps and for the second stage the M version[7] with 50 inference steps. We use both the DDPMS scheduler and guidance scale of 7 for the first stage and 4 for the second stage. With this configuration, we produce images of size 256×256.

8. Semantic Link

To investigate the impact of semantic relationships between objects, we select 28 COCO labels from three macro-classes, *vehicles, animals, and foods*, and generate images using a template with 2 objects. In order to study the influence of semantic links between objects with the same prompt, we consider the following dissimilarity metric on the set $\mathcal{O}$.

$$\forall o_x, o_y \in \mathcal{O}, o_x \neq o_y,$$
$$d(o_x, o_y) = d(o_y, o_x) = \frac{\text{TIAM}_{z_i} + \text{TIAM}_{z_j}}{2} \quad (2)$$

$$\forall o_x \in \mathcal{O}, \qquad d(o_x, o_x) = 0 \quad (3)$$

where z_i is (o_x, o_y) and z_j is (o_y, o_x). For TIAM$_z$ we compute the score per z (*i.e.* per prompt). Using this dissimilarity between the labels, we project them with Multidimensional Scaling (MDS) into a 2D space, as represented in Fig. 7 for SD 1.4, in Fig. 8 for SD2, in Fig. 9 for unCLIP and in Fig. 10 for IF. The projection can be interpreted such that the closer two labels are, the more challenging it becomes for the model to represent them together. Note that we obtained similar projections with t-SNE instead of MDS.

For all the models, we observe some clusters of objects from the same macro-class, in particular *animals*. It shows that **when two objects are semantically close, they tend to be harder being generated in the same image**. Tang *et al*. [18] obtain results in the same vein, showing that it is easier to generate a non-cohyponym than a cohyponym.

[2] https://huggingface.co/CompVis/stable-diffusion-v1-4
[3] https://huggingface.co/stabilityai/stable-diffusion-2-base
[4] https://huggingface.co/docs/diffusers/api/schedulers/multistep_dpm_solver
[5] https://huggingface.co/kakaobrain/karlo-v1-alpha
[6] https://huggingface.co/DeepFloyd/IF-I-L-v1.0
[7] https://huggingface.co/DeepFloyd/IF-II-M-v1.0

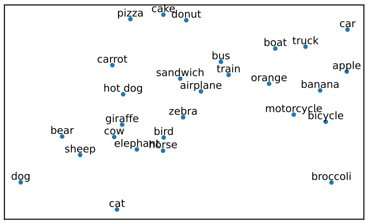


Figure 7. MDS on the objects score dissimilarity for SD 1.4.

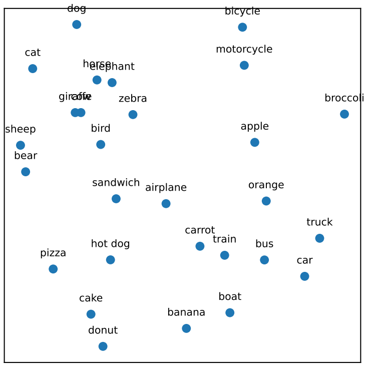


Figure 8. MDS on the objects score dissimilarity for SD 2.

However, the effect remains slight, suggesting that the cohyponymy has either an indirect or minor link to this difficulty of generation. We quantified the effect by computing the correlation between the TIAM score for all the templates with two objects and the semantic distance between the two objects. We used various methods to estimate the semantic distance, including Wu-Palmer, the CLIPscore, and the cosine similarity between the embedding of the token in the prompt (before the attention of the transformer) for SD 1.4 and SD 2. For all distances, these correlations were negative (confirming the effect) but their absolute values were less than 0.5 (confirming the effect is slight).

9. Attribute Binding

We report the TIAM per object in Fig. 11, showing that the first object is more often generated and correctly colored. We compute the *binding success rate*, but by differentiating by colors for attributes in the first position (Fig. 12) and attributes in the second position (Fig. 13). We observed that the models face greater difficulty in assigning green and blue colors when two objects are involved (parallel with a single label case). It is worth noting that IF performs better than other models.

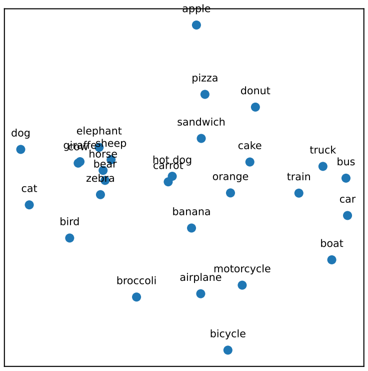


Figure 9. MDS on the objects score dissimilarity for unCLIP.

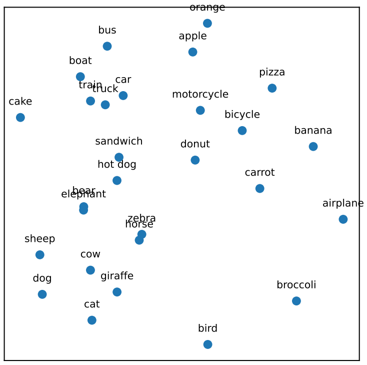


Figure 10. MDS on the objects score dissimilarity for IF.

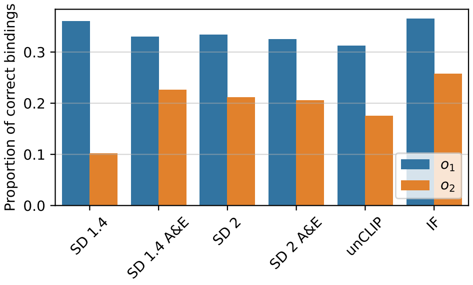


Figure 11. TIAM per object *i.e.* proportion of correct generated object with the correct binding.

10. Latent Diffusion Model

We remind the architecture of the Latent Diffusion Model of Rombach *et al.* (2021) in Fig. 14, since the theoretical explanation of Section 4.2 of the main paper relies on it.

During the investigation of the seed performances of the models, we made a noteworthy finding. We observed that the SD models exhibited similar behavior in relation to the

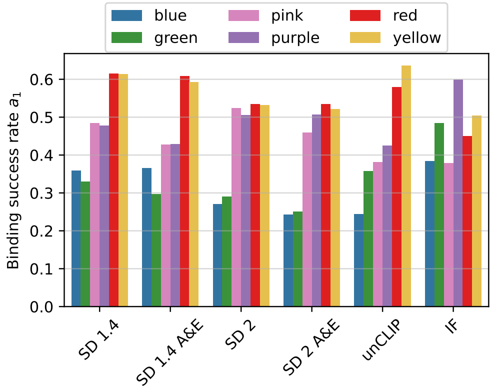


Figure 12. Binding success rate for the first object, among the first objects correctly detected.

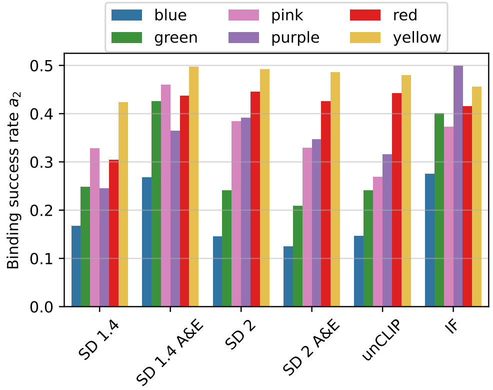


Figure 13. Binding success rate for the second object, among the second objects correctly detected.

seeds (if we do not consider the performance gap between the score of the model) *i.e.* when we standardize the score of each seed for each model (Fig. 15) the score exhibits a remarkably similar trend for both models with the same *"good"* and "*bad*" seeds. However, we explain in the article (Section 4.2) that the *"good"* and "*bad*" seeds are specific to each model, which may seem contradictory at first glance.

To explain this point, we need to remember the training of diffusion models. Let x_0 be an original image. We had on the image a scaled quantity of noise $\epsilon \sim \mathcal{N}(0, 1)$ to obtain x_t ($\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$) and the U-net ϵ_θ try to predict with the added noise (the loss function ($\mathbb{E}_{\mathcal{E}(I),\epsilon\sim\mathcal{N}(0,1)}\left[||\epsilon - \epsilon_\theta(x_t, t)||_2^2\right]$). At each training step a t is drawn the model must predict the noise. In the case of LDM, just replace the x with z because the diffusion process is in the latent space.

As the two models are trained on the same data, we hypothesize that they are trained on the same z_t *i.e.* they are trained to predict the same ϵ. However ϵ is random, we suspect that all the z_t are precomputed and the associate ϵ saved, and they used the same for the training of both models. Because of this particularity during the training, the models try to retrieve the z_0 from multiple possible t. The models learn a similar path of reverse diffusion. At inference, when we draw a similar noise χ_T, the reverse process will be similar and conduct to near χ_t and finally to a near χ_0. We show in Fig. 16 that with the same prompt and the same starting noise, we exhibit strong composition similarity that explains the parallel performance of the seed.

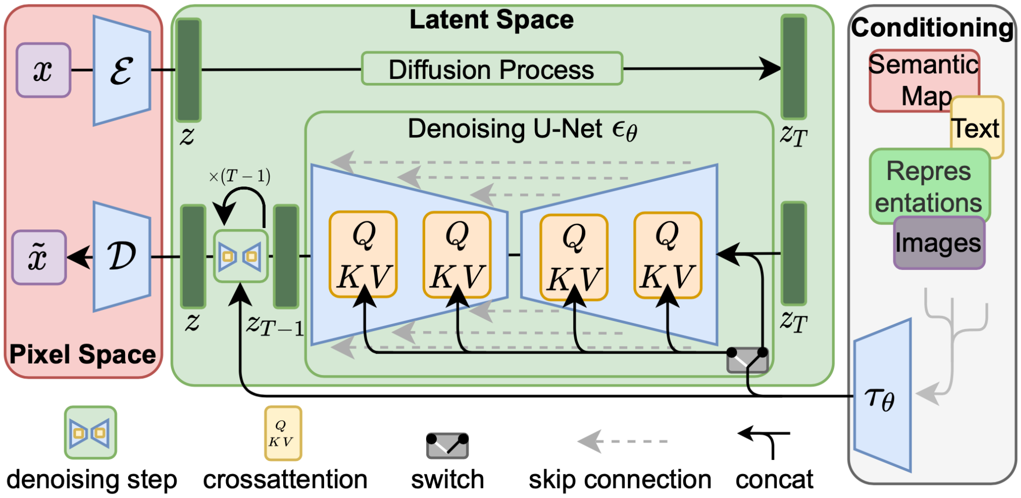


Figure 14. Architecture of the Latent Diffusion Model [16].

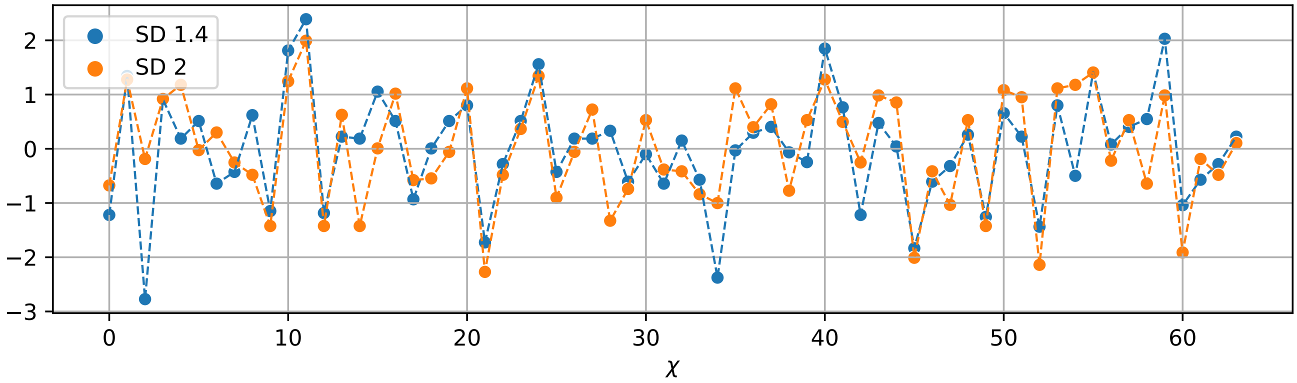


Figure 15. Standardized score per seed for SD 1.4 and SD 2. We observe that both of them have globally the same "good" and "bad" seeds (see Section 10 for explanations).

11. Human Evaluation and Comparison to Other Automatic Metrics

We conducted a human evaluation to determine how much TIAM is aligned to it, and compared it to two other automatic metrics based on the CLIP [15] score and BLIP [12] score.

We randomly sample 32 prompts, comprising 16 with one object and one attribute/color (referred to as $\mathcal{C}$), and 16 with two objects (referred to as $\mathcal{O}$). Subsequently, we randomly select one image per prompt and record the corresponding scores provided by our metrics for each image. We use images generated by the IF model. We then solicit human evaluators to discern whether they perceive alignment between the given prompt (utilized for generation) and the associated image. An illustrative example is given in Fig. 17. For a fair comparison with CLIP and BLIP, which solely yield a similarity score between images and captions, we refrained from rephrasing the prompts into specific questions like ”Is the first object present?” or ”Is the second object present?”, that would have given an advantage to TIAM.

We conducted our study on 57 humans aged from 7 to 79. Only 6 of them could be considered text-to-image (T2I) experts (the author who made the study did not participate in the assessment), while other human subjects never manipulated T2I models, or even didn't know it could exist. In any case, the agreement of TIAM with the experts' assessment was not significantly different than that with non-experts. Nor did we find any significant difference in terms of gender or age. For non-English speakers, the prompts were translated into their native language (in particular for

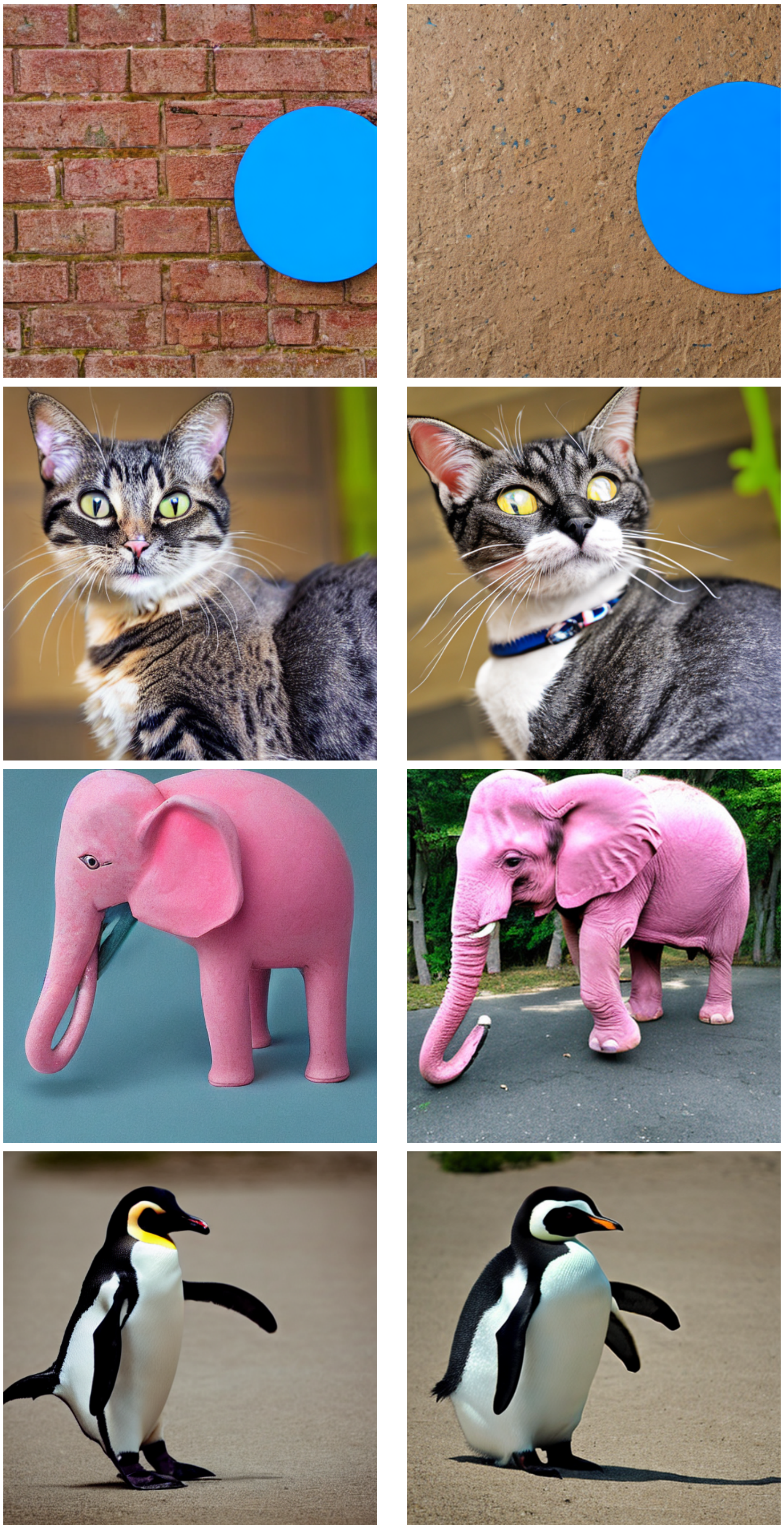
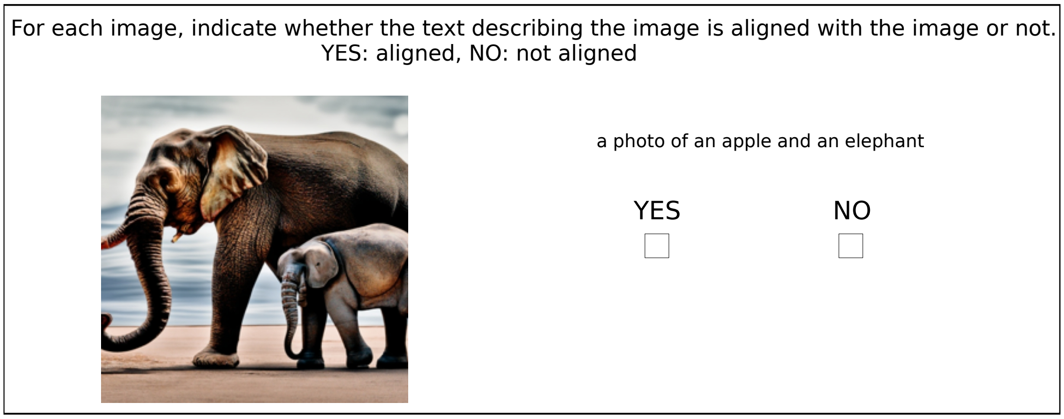


Figure 16. We present images obtained with the same seed and same prompt for SD 1.4 (left) and SD 2 (right). Note how similar the compositions are. The prompt and seed are respectively ”*a photo of a blue circle*” seed 17, ”*a photo of a cat*” seed 27, ”*a photo of a pink elephant*” seed 18, and ”*a photo of a penguin*” seed 45.



For each image, indicate whether the text describing the image is aligned with the image or not.
YES: aligned, NO: not aligned

a photo of an apple and an elephant

YES □ NO □

Figure 17. Extract from the study for the human evaluation.

subjects less than 15 years old).

For automatic methods, we used the OpenAI ViT-B/32 CLIP model [8] and the Salesforce BLIP model [9]. The BLIP and CLIP score is a similarity score between the embedding of the image and the caption used to generate the image.

The agreement between the human annotator was assessed in terms of Fleiss' kappa [4]. According to Landis and Koch [10], a kappa of $[0.21 - 0.40]$ is *fair*, that in $[0.41 - 0.60]$ is *moderate*, that in $[0.61 - 0.80]$ is *substantial* and the agreement is *almost perfect* when the Fleiss' kappa is in $[0.81 - 1.00]$. With a global value of $\kappa = 0.73$ the agreement of the human annotators of our study is thus *substantial*. If one distinguishes the two subsets, the agreement on $\mathcal{C}$ is in the upper range of the *moderate* agreement (0.59) while that for $\mathcal{O}$ is *almost perfect* (0.85). It nevertheless shows that, even for humans, characterizing the colors of an object may be an ambiguous task. We illustrate for instance the example that led to the most disagreement between annotators in Fig. 18.

We compute the Pearson correlation between human decisions and TIAM, the CLIP score, and the BLIP score. The results are reported in Tab. 4. TIAM exhibits a significantly stronger correlation with human judgments compared with other metrics. We have a similar conclusion if the alignment is estimated with the Spearman'rank correlation (Tab. 5).

Lastly, we would like to emphasize that TIAM captures the model's success rate more comprehensibly in contrast to other automatic similarity scores. While CLIP and BLIP can serve to compare two generative models (*e.g.* evaluate models based on their CLIP score) their inherent meaning is limited. We note also that CLIP has a poor compositional understanding, limiting a precise evaluation of text-image alignment [21]. In addition, TIAM enables the analysis of specific modalities, yielding insightful outcomes such as the success rate per seed and the proportion of apparition of an object according to its position in the prompt.

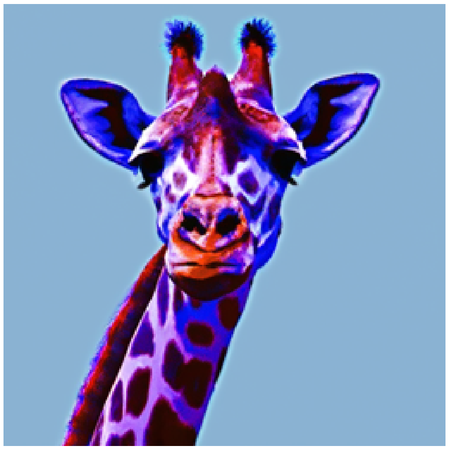


Figure 18. "A photo of a blue giraffe" generated with IF. The human annotators had quite low agreement on the alignment of this image with the prompt.

Score	$\mathcal{C}+\mathcal{O}$	$\mathcal{C}$	$\mathcal{O}$
CLIP	$0.47_{p=6\times10^{-3}}$	$0.22_{p=4\times10^{-1}}$	$0.62_{p=1\times10^{-2}}$
BLIP	$0.67_{p=3\times10^{-5}}$	$0.48_{p=6\times10^{-2}}$	$0.77_{p=5\times10^{-4}}$
TIAM	$\mathbf{0.82}_{p=7\times10^{-9}}$	$\mathbf{0.70}_{p=2\times10^{-3}}$	$\mathbf{0.98}_{p=2\times10^{-11}}$

Table 4. Pearson correlation between human decisions and TIAM/BLIP/CLIP. $\mathcal{C}+\mathcal{O}$ stands for the correlation without distinction of the series. p is the p-value for the null hypothesis H_0: *the distributions underlying the samples are uncorrelated*, the alternative hypothesis is *the correlation is non zero*.

[8] https://huggingface.co/openai/clip-vit-base-patch32

[9] https://huggingface.co/Salesforce/blip-itm-base-coco

Score	$\mathcal{C} + \mathcal{O}$	$\mathcal{C}$	$\mathcal{O}$
CLIP	$0.50_{p=2\times10^{-3}}$	$0.39_{p=7\times10^{-2}}$	$0.53_{p=2\times10^{-02}}$
BLIP	$0.38_{p=2\times10^{-2}}$	$0.06_{p=4\times10^{-1}}$	$0.64_{p=4\times10^{-3}}$
TIAM	$\mathbf{0.82}_{p=1\times10^{-4}}$	$\mathbf{0.77}_{p=1\times10^{-3}}$	$\mathbf{0.87}_{p=2\times10^{-4}}$

Table 5. Spearman correlation between human decisions and TIAM/BLIP/CLIP. $\mathcal{C} + \mathcal{O}$ stands for the correlation without distinction of the series. p is the p-value for the permutation test.

12. Scalability of TIAM

In this section, we initially address strategies to manage the potential growing complexity inherent in the template approach. Subsequently, we explore methods to go beyond the limited set of COCO labels.

12.1. Scalability

The template approach can become cumbersome when dealing with multiple modalities (*e.g*. exploring a prompt template with 5 objects using 30 different objects ($/| = 30$) leads to 17 100 720 prompts).

To alleviate this complexity, we can adopt a sampling approach, consisting of randomly drawing a defined number of prompts to estimate the results (*e.g*. TIAM score, the proportion of occurrence of the object according to its position in the template, . . .). We conducted such a study on several experiments reported in the paper : (A) the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1), (B) the prompts with 2 objects created with the combination of 28 COCO labels (Section 4.3, part on the semantic link), and (C) the prompts with 2 objects and associated attribute (Section 4.4).

Using all generated prompts, at each step, we draw (without replacement) n prompts and we compute the results from these samples. We start with 50 prompts and increase n up to the maximum number (all possible prompts) by a step of 2 (thus using $52, 54, 56...$ samples). We made this for each model used for the respective experiments. For (A) and (B) we report the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed, for (C) we report the TIAM score, the proportion of occurrences of each object based on its position in the prompt and the success rate of color attribution w.r.t the detected objects. Hence we report below the results for:

- SD 1-4 (A) Fig. 19 (B) Fig. 25 (C) Fig. 29,
- SD 1.4 A&E (A) Fig. 20 (C) Fig. 30,
- SD 2 (A) Fig. 21 (B) Fig. 26 (C) Fig. 31,
- SD 2 A&E (A) Fig. 22 (C) Fig. 32,
- IF (A) Fig. 23 (B) Fig. 27 (C) Fig. 33,
- unCLIP (A) Fig. 24 (B) Fig. 28 (C) Fig. 34.

Across all our findings, a marked trend emerges from around 300 prompts, indicating the viability of employing a sampling method to approximate the results presented in the main study and alleviate the complexity of the template-based approach of TIAM in practice.

In addition, modal-specific studies can be conducted. For instance, to study certain modalities, we can imagine isolating each modality for individual scrutiny such as defining a few potential objects, but above all varying the modality we wish to study. This approach was applied to the attribute binding in Section 4.4 by reducing the number of objects studied and prioritizing the exploration of attribute binding.

Finally, we would like to emphasize that, when exploring semantic links comprehensively, it is essential to test the effect of each word placed together (Section 4.3).

12.2. TIAM with other labels

TIAM does not depend on the COCO labels and can be applied to other labels as long as we have a detector capable of detecting the desired studied labels. Indeed, TIAM can be implemented with any other detection model, trained on other labels, to evaluate the prompt-image alignment of the T2I models. In particular, the open-vocabulary detection models field has emerged (*e.g*. for detection [6, 13, 20] segmentation [5, 11]) presenting itself as a robust contender for surpassing the limitations imposed by constrained label sets.

12.3. TIAM with other attributes

See Section 3 of the Supplementary Material for a discussion on how to consider other attributes than *colors*, such as *size* or *texture*.

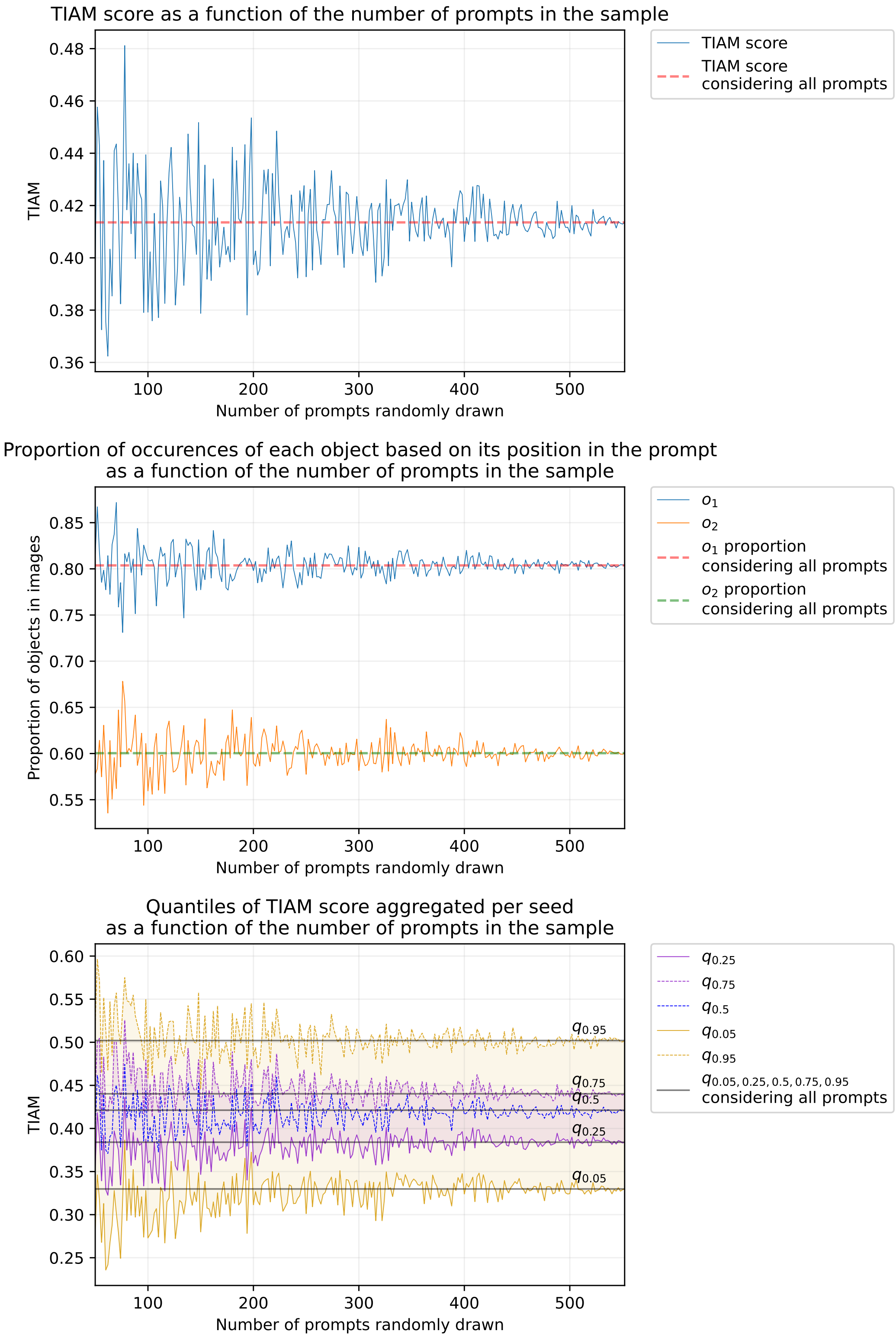


Figure 19. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 1.4, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

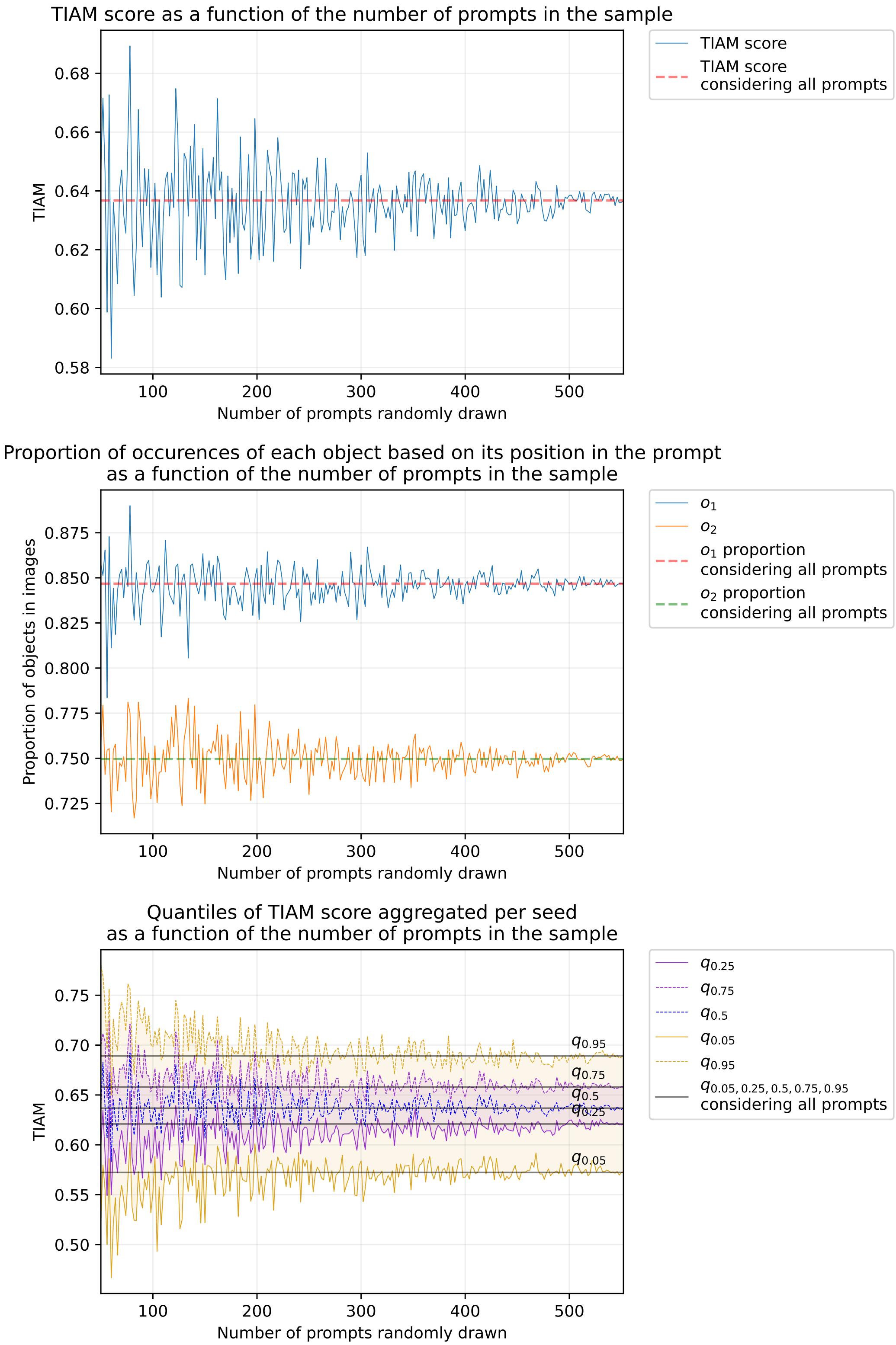


Figure 20. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 1.4 A&E, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

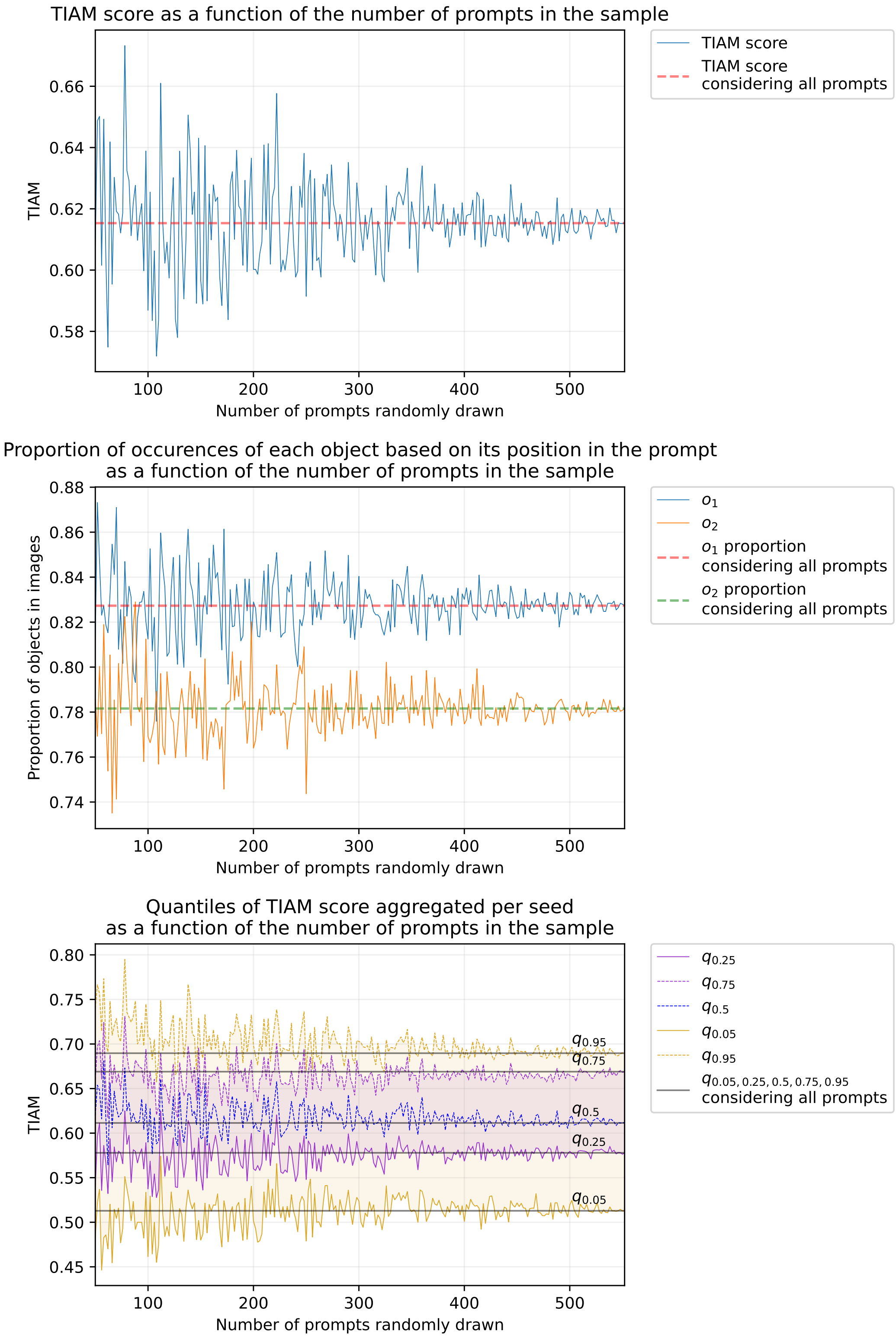


Figure 21. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 2, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

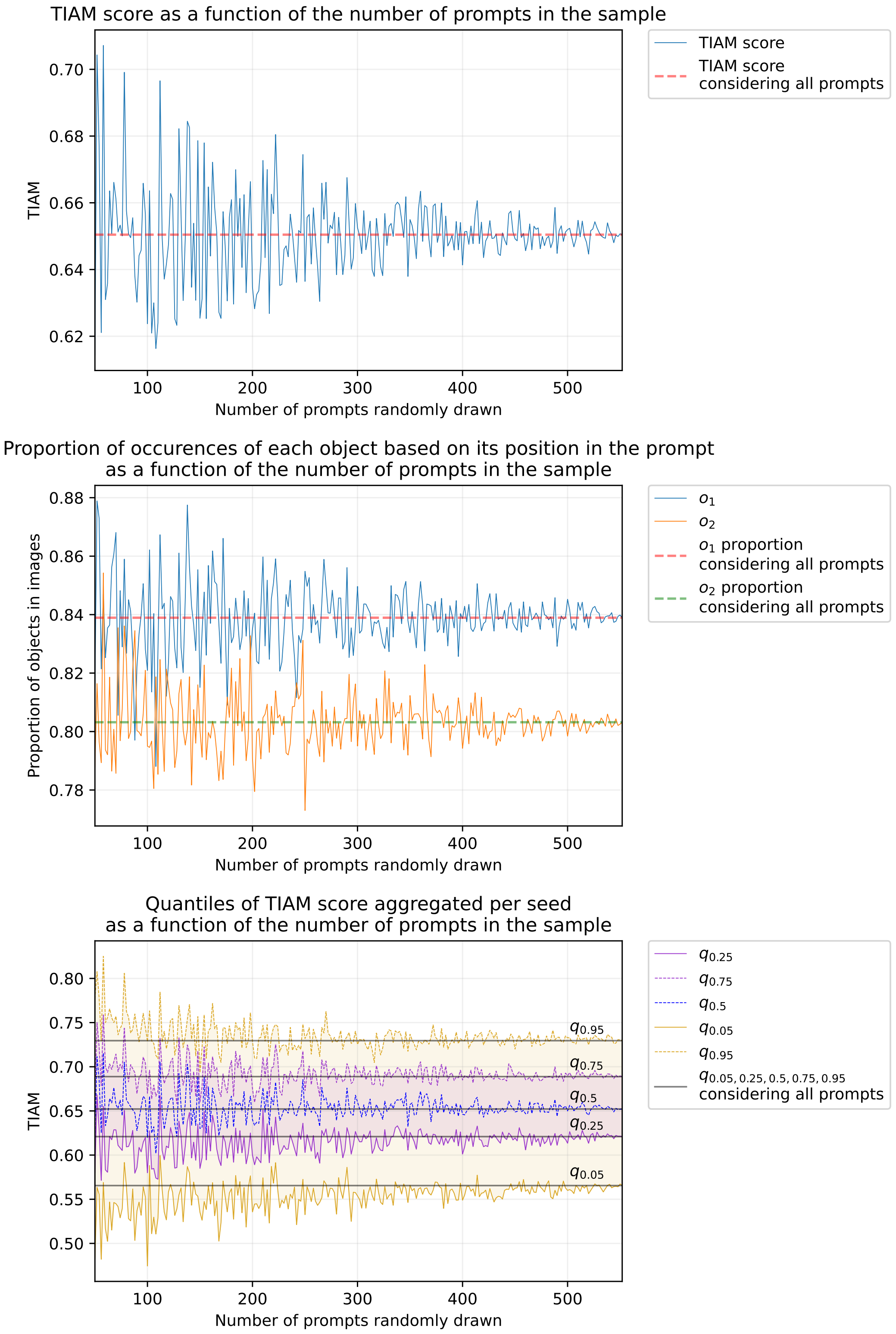


Figure 22. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 2 A&E, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

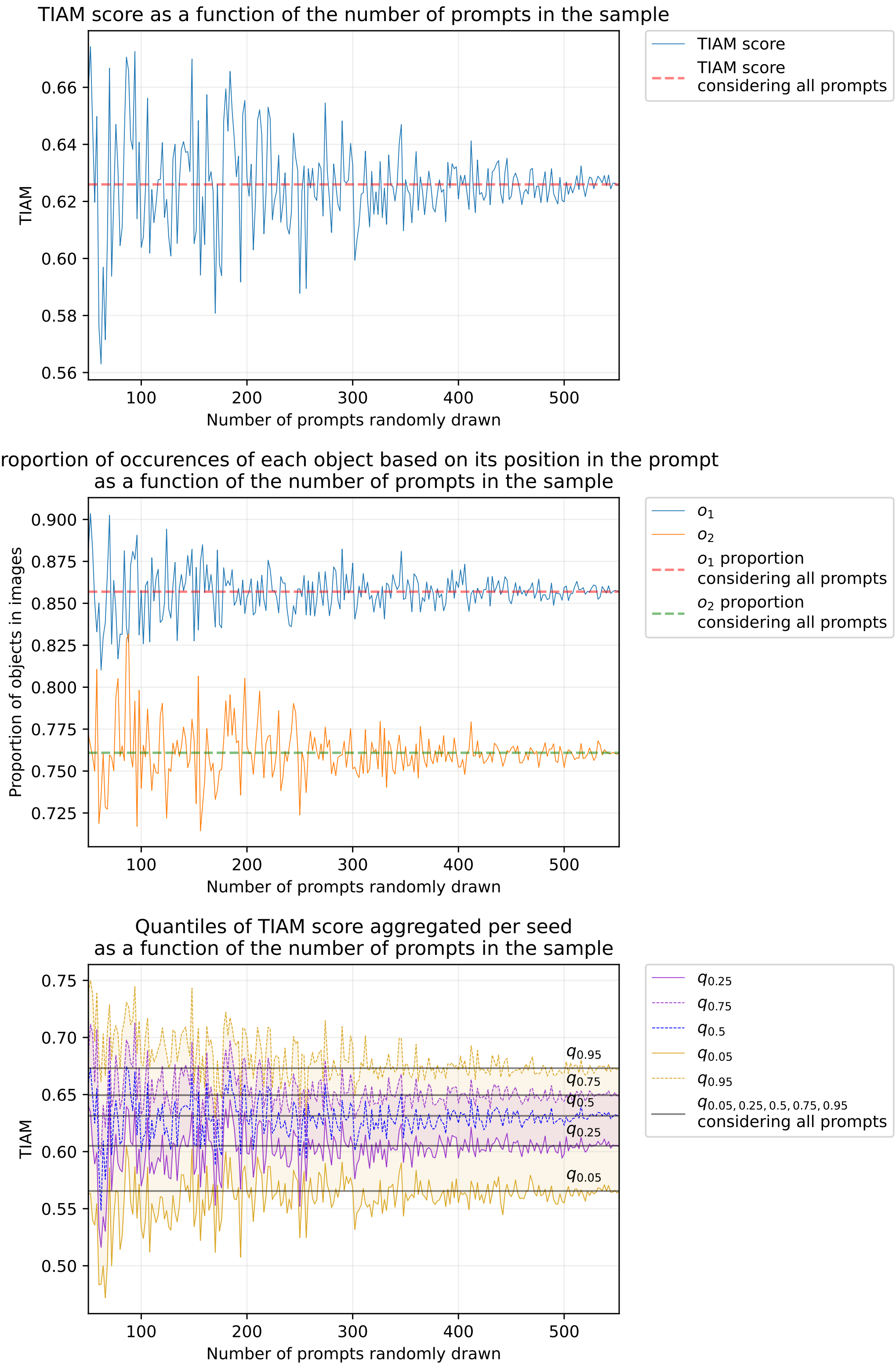


Figure 23. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for IF, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

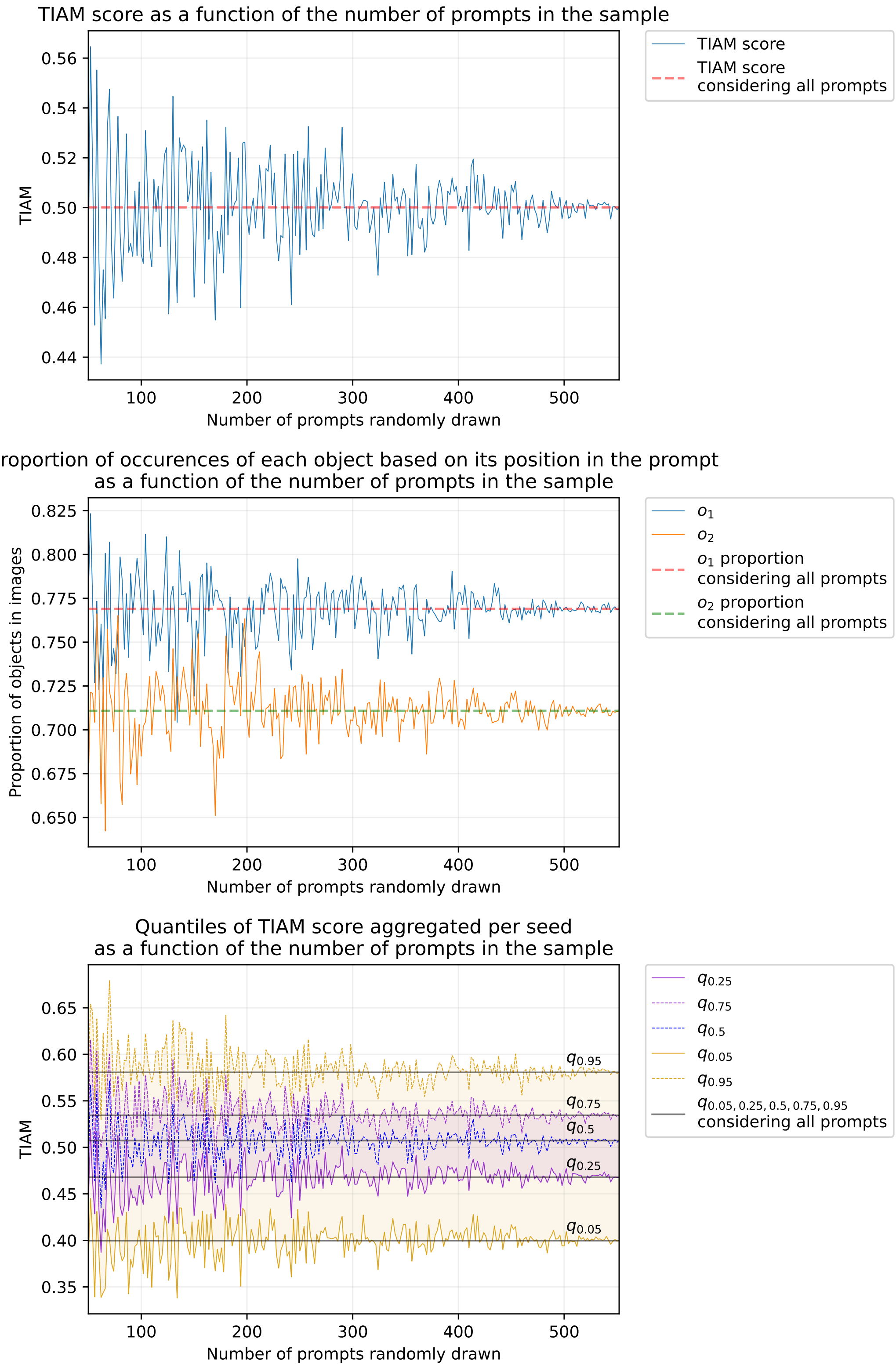


Figure 24. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for the unCLIP, using the prompts with 2 objects created with the combination of 24 COCO labels (Section 4.1).

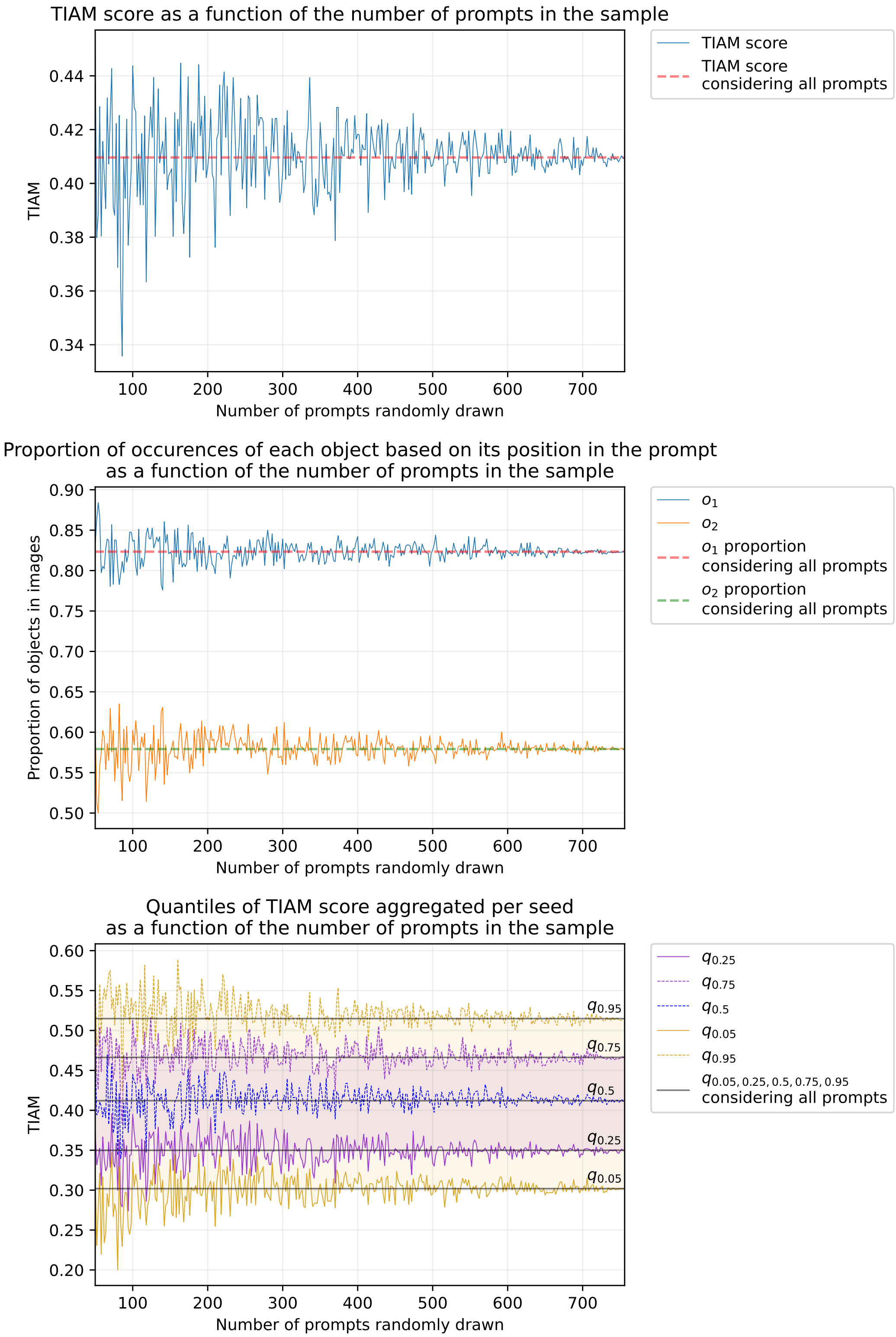


Figure 25. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 1.4, using the prompts with 2 objects created with the combination of 28 COCO labels (Section 4.3, part on the semantic link).

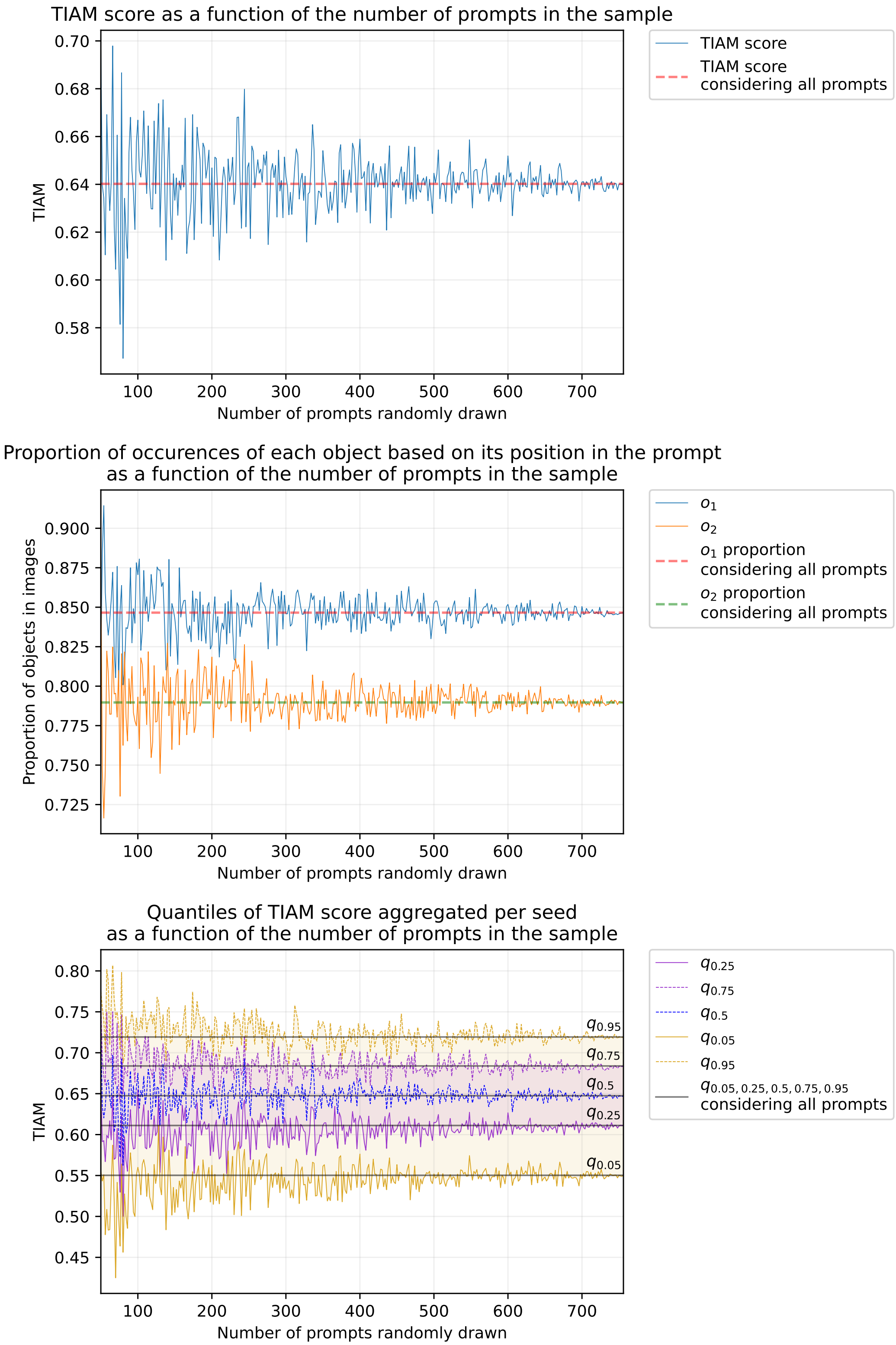


Figure 26. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for SD 2, using the prompts with 2 objects created with the combination of 28 COCO labels (Section 4.3, part on the semantic link).

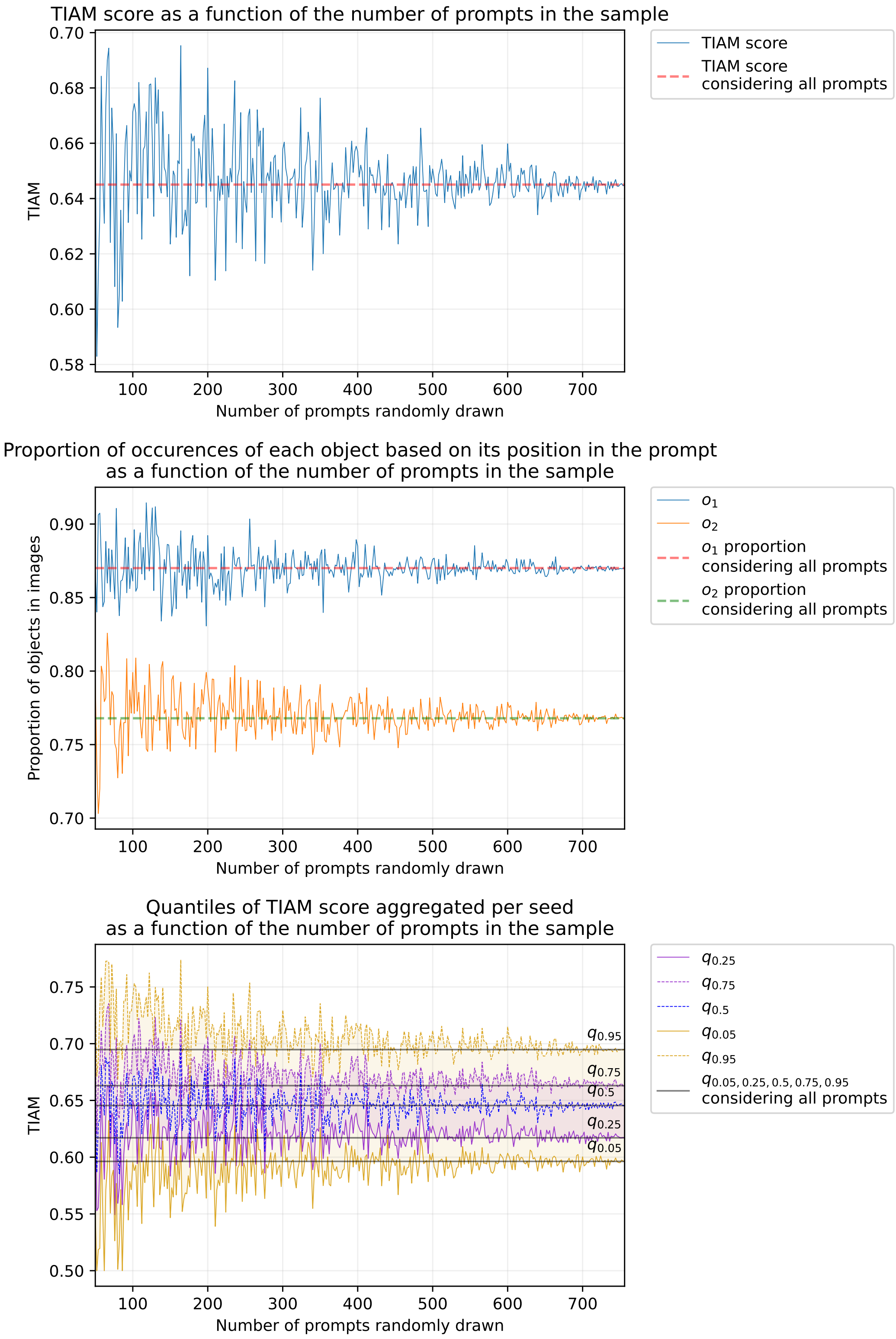


Figure 27. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for IF, using the prompts with 2 objects created with the combination of 28 COCO labels (Section 4.3, part on the semantic link).

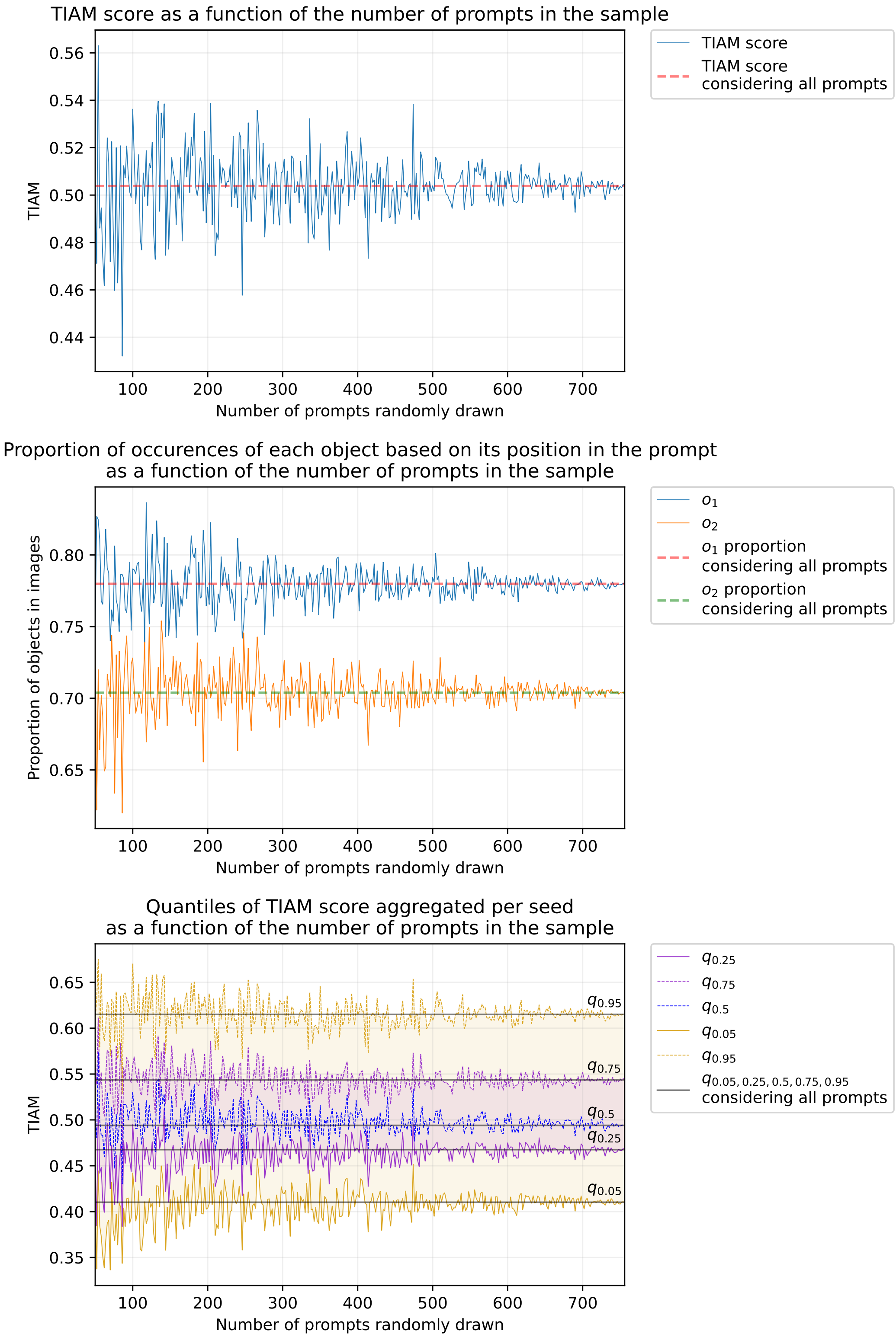


Figure 28. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the quantiles of the TIAM score aggregated per seed as a function of the number of prompts randomly drawn to compute the results, for unCLIP, using the prompts with 2 objects created with the combination of 28 COCO labels (Section 4.3, part on the semantic link).

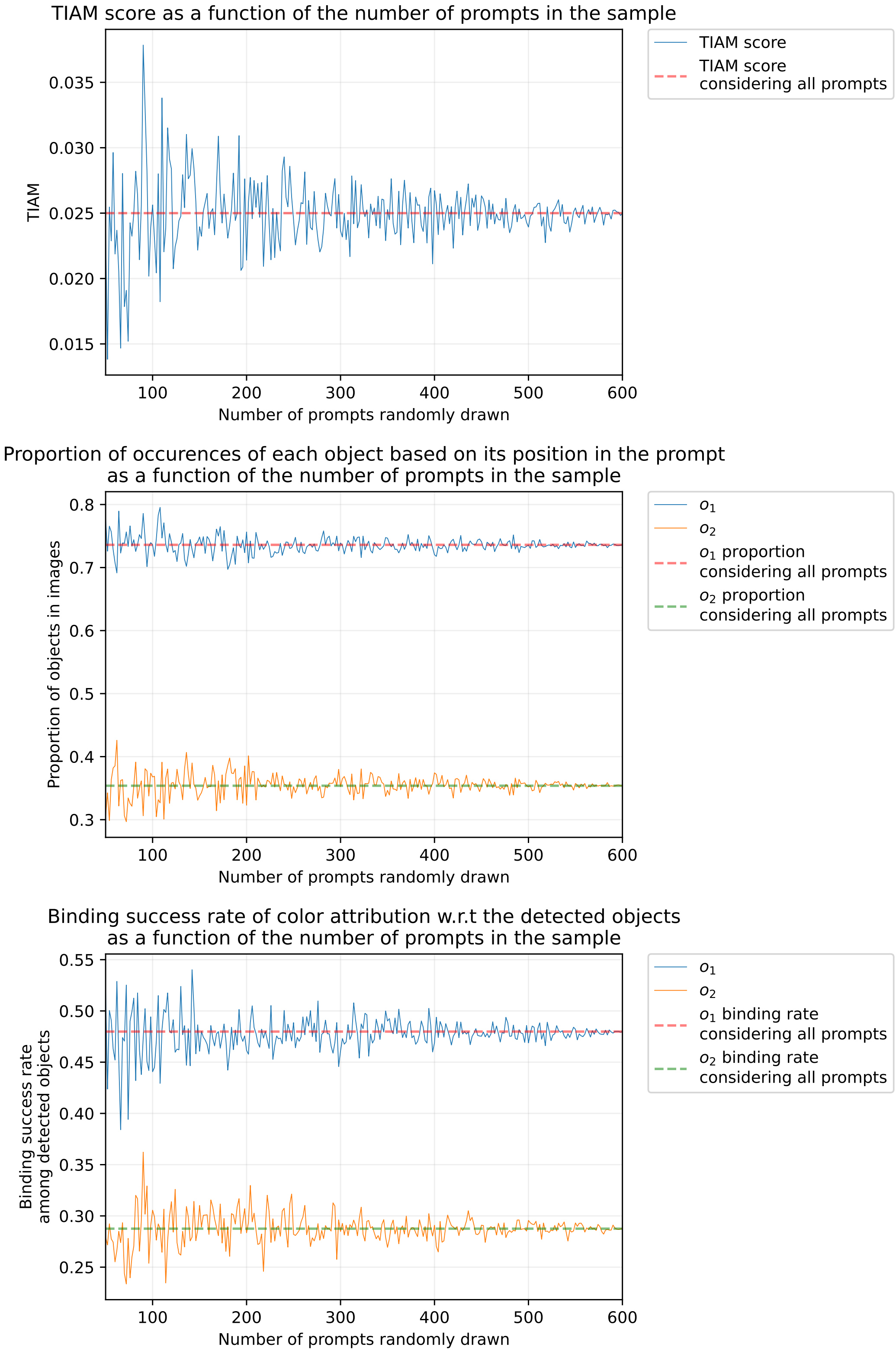


Figure 29. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for SD 1.4, using the prompts with 2 objects and associated attribute (Section 4.4).

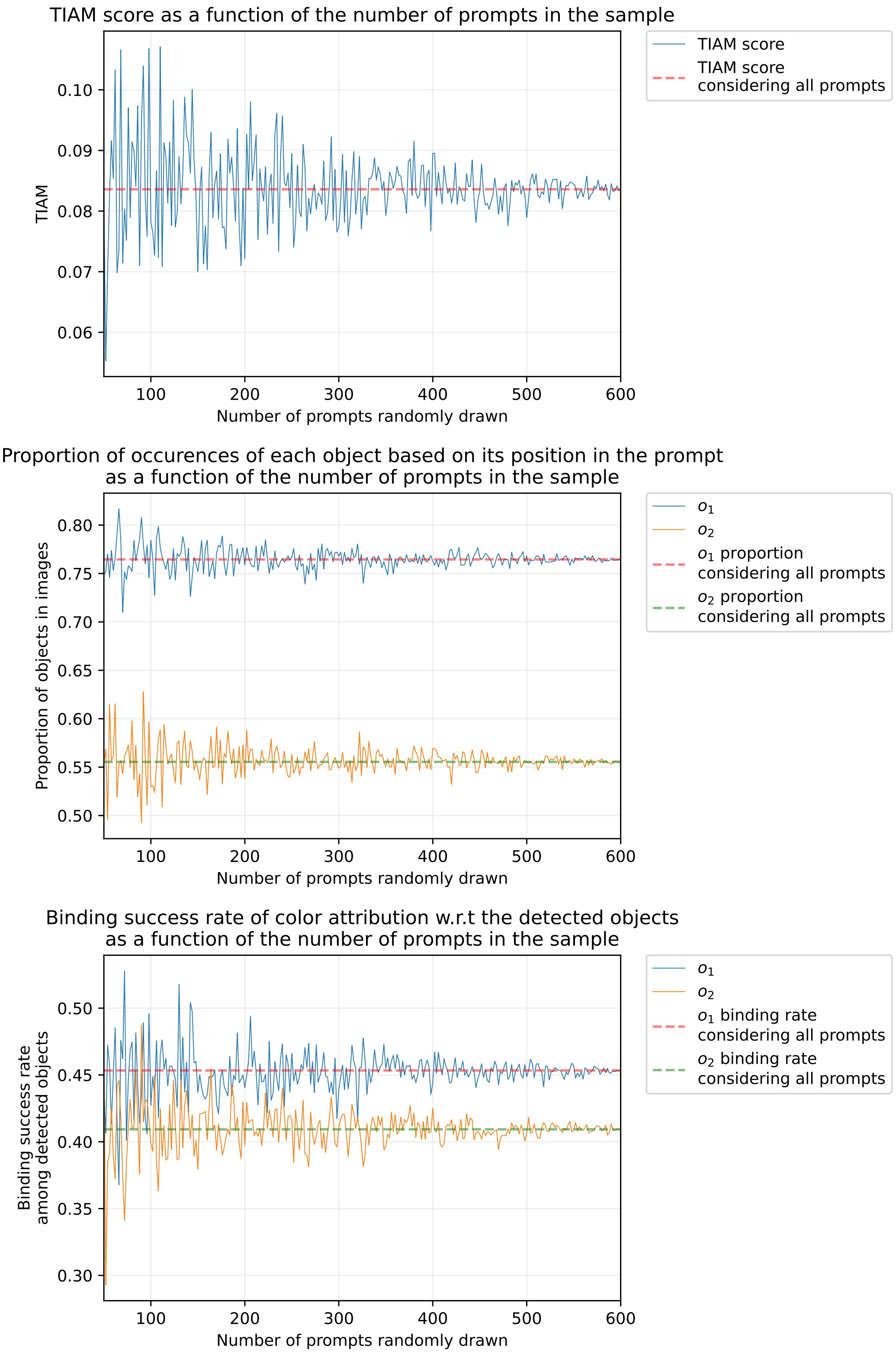


Figure 30. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for SD 1.4 A&E, using the prompts with 2 objects and associated attribute (Section 4.4).

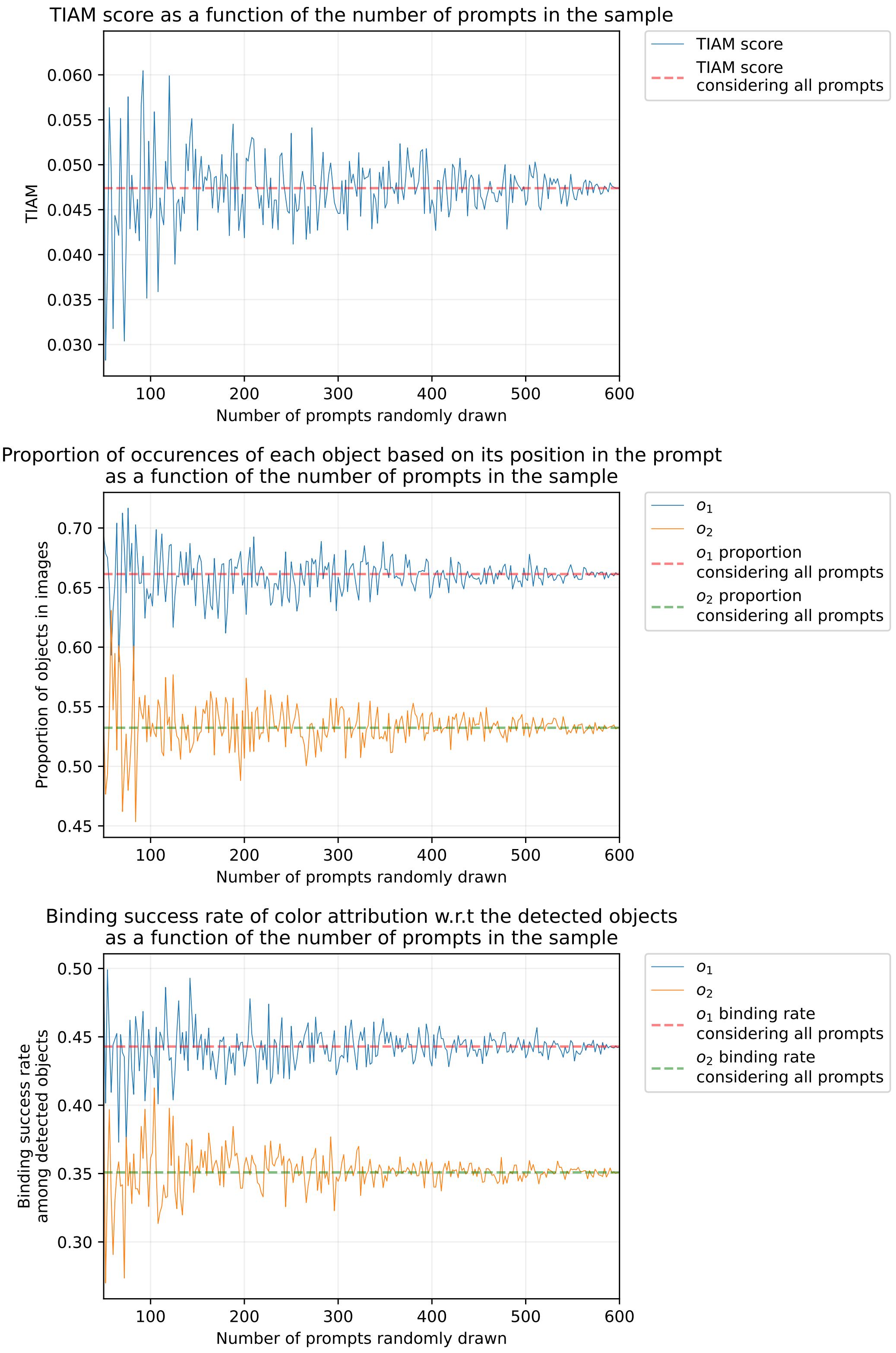


Figure 31. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for SD 2, using the prompts with 2 objects and associated attribute (Section 4.4).

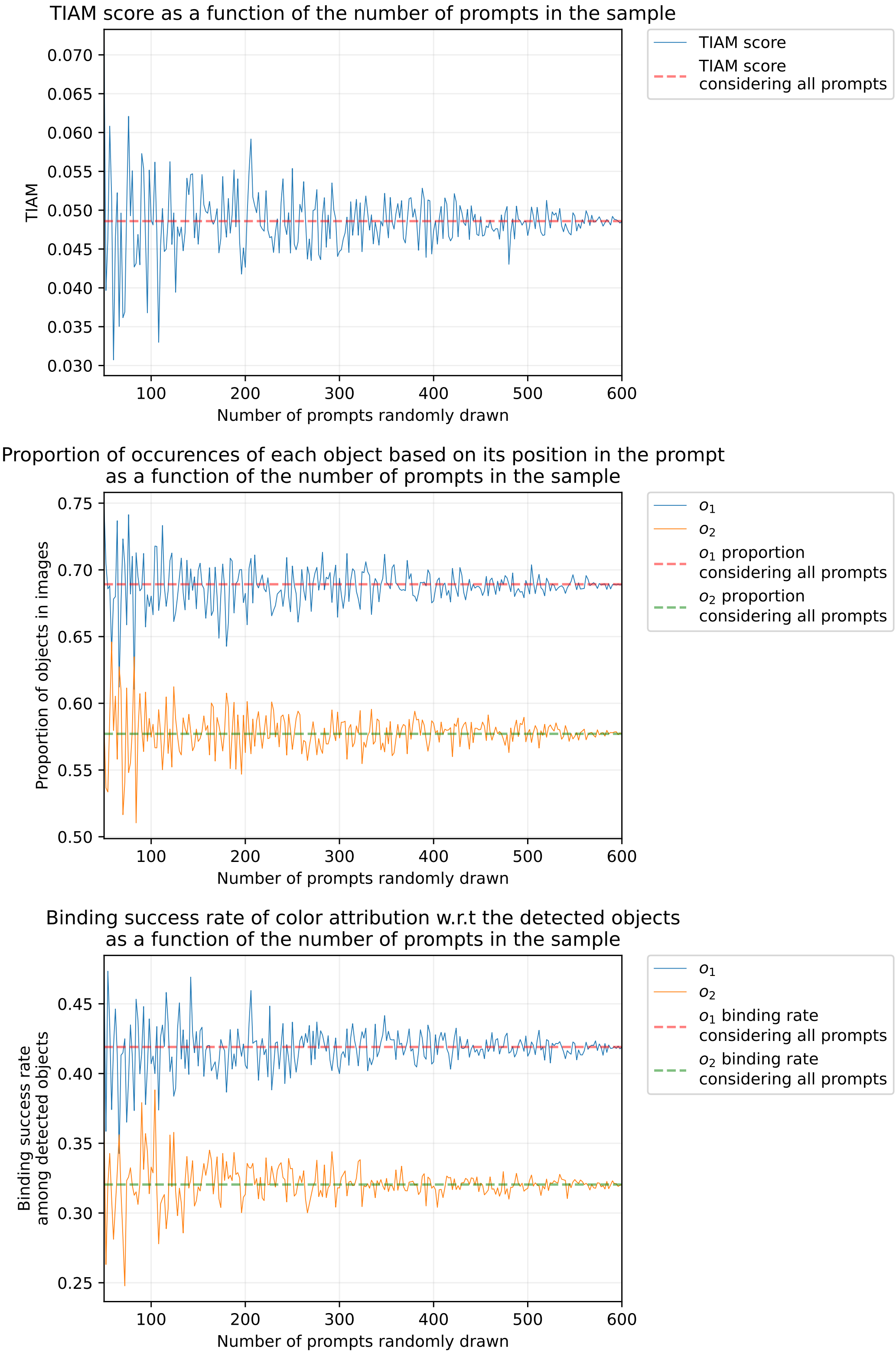


Figure 32. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for SD 2 A&E, using the prompts with 2 objects and associated attribute (Section 4.4).

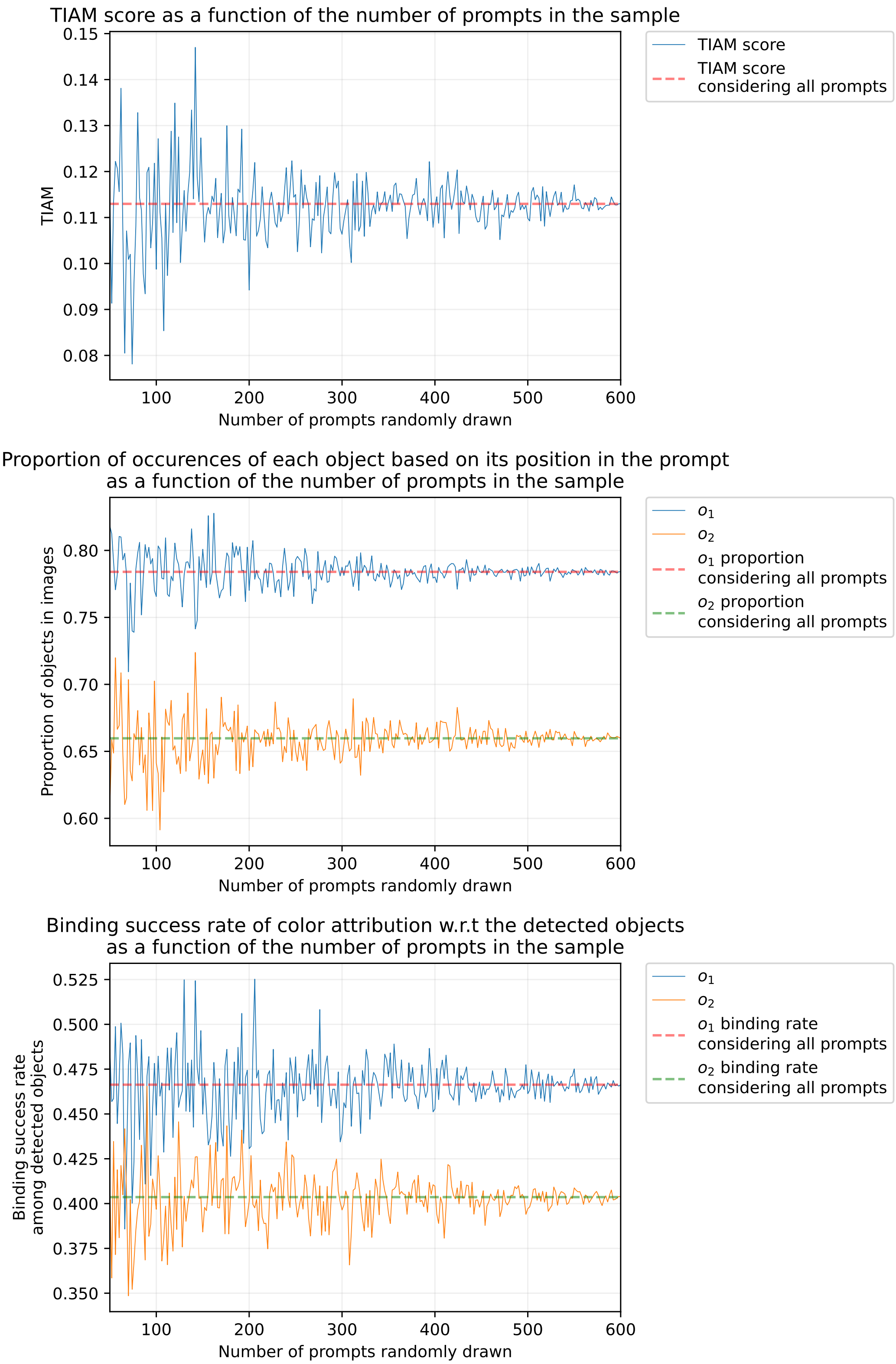


Figure 33. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for IF, using the prompts with 2 objects and associated attribute (Section 4.4).

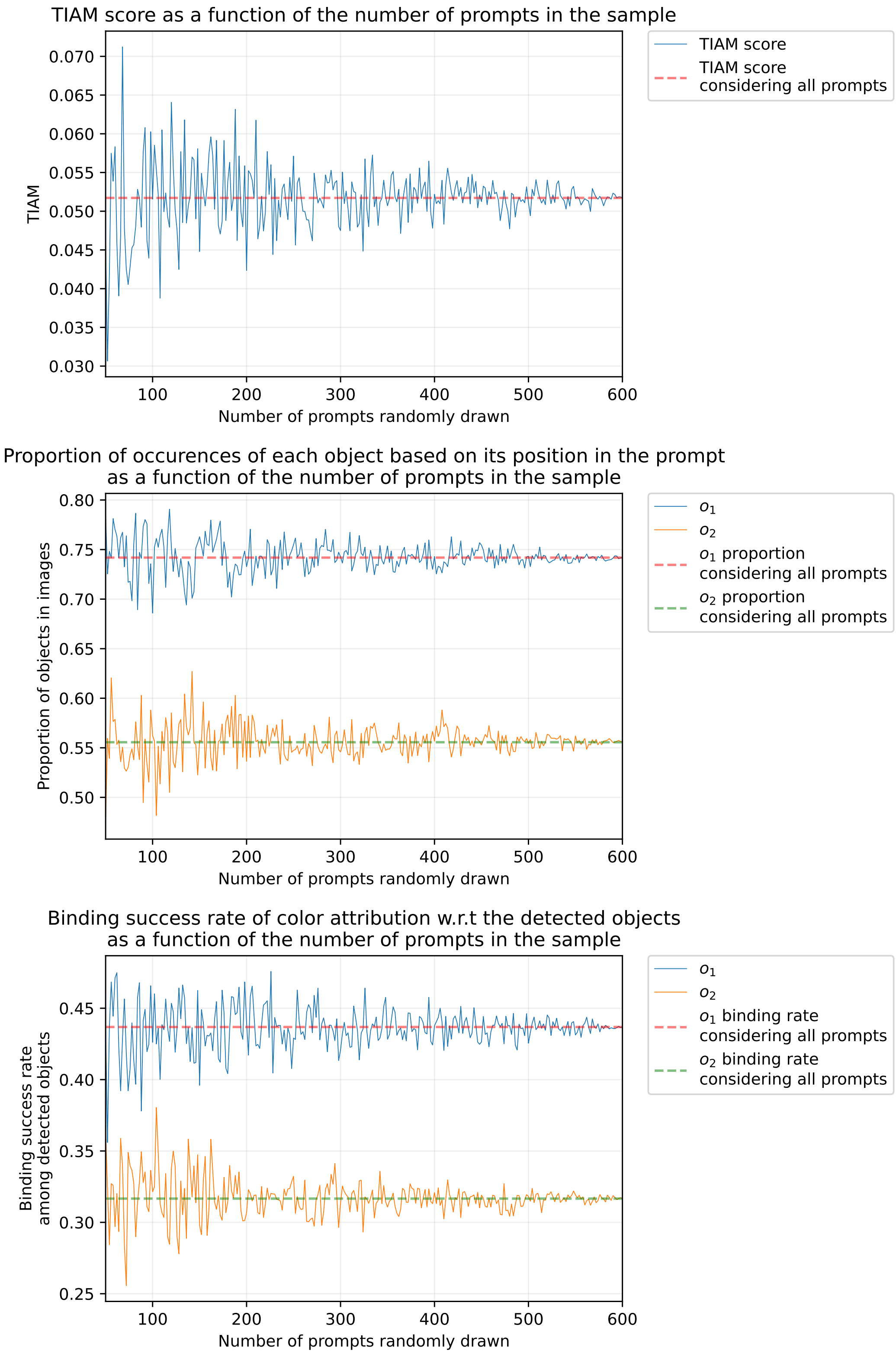


Figure 34. Evolution of, respectively, the TIAM score, the proportion of occurrences of each object based on its position in the prompt, and the success rate of color attribution w.r.t the detected objects as a function of the number of prompts randomly drawn to compute the results, for unCLIP, using the prompts with 2 objects and associated attribute (Section 4.4).

References

[1] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, Los Angeles, 1969.

[2] N. Bhushan, A. R. Rao, and G. L. Lohse. The texture lexicon: Understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science*, 21(2):219–246, 1997.

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[4] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378—-382, 1971.

[5] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 540–557, Cham, 2022. Springer Nature Switzerland.

[6] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv 2104.13921*, 2022.

[7] Humeau-Heurtier. Texture feature extraction methods: A survey. *IEEE Access*, 7:8975–9000, 2019.

[8] Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8(2), 1962.

[9] Paul Kay and Richard S. Cook. *World Color Survey*, pages 1265–1271. Springer New York, New York, NY, 2016.

[10] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977.

[11] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.

[12] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[13] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022.

[14] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan LI, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, pages 5775–5787. Curran Associates, Inc., 2022.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[17] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J. Fleet. Monocular depth estimation using diffusion models. *arXiv 2302.14816*, 2023.

[18] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5644–5659, Toronto, Canada, July 2023. Association for Computational Linguistics.

[19] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[20] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7031–7040, June 2023.

[21] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023.