

464 A LVLMs-as-classifiers prompts

465 **BDD100K Dataset** The prompt for LVLM-based classification is provided in Table 6, considering
 466 'start' and 'stop' as the {str_categories}. The LVLM is forced to focus on the semantics that define
 467 each driving situation, since they are definitive for classification based on concepts.

Table 6: Classification prompt for BDD100K

Classify each image in their appropriate class according to the driving situation they depict. Valid class labels are {str_categories} and only these, depending on whether the car has to move or stop based on its surroundings. You need to classify the images in one of these classes. Pay attention to the semantics that define each class. Return me only the label of the scene depicted and nothing else.

468 **Visual Genome Dataset** In Table 7 we show the classification prompt used to classify an image
 469 from Visual Genome in one of its appropriate categories belonging in the {str_categories} list.
 470 The LVLM is forced to focus on the semantics that define each class, since they are definitive for
 471 classification based on concepts.

Table 7: Classification prompt for Visual Genome

Classify each image in their appropriate class according to the scene they depict. Valid classes are {str_categories} and only these, so you need to classify the images in one of these classes. Pay attention to the semantics that define each class. Return me only the label of the scene depicted and nothing else.

472 B Prompts for performing the edits

473 As mentioned in the 3.2 section, there are three ways to define which edits are going to be performed
 474 and in which order.

475 In the *local editing* approach, the LVLM serves as the only decision-making module to order the
 476 edits produced from the explanation component. In each step, only one edit is selected and passed to
 477 the generative component. This assists in performing a small number of steps until label flip, since
 478 label flip may occur before the edits proposed in E is exhausted (an assumption that is verified, based
 479 on the results of Table 4, in which our generative V-CECE consistently performs fewer edits than
 480 its non-generative counterparts). Other than that, performing one step at a time allows for more
 481 high-quality generations from the point of the generative component.

482 The prompt that arranges the local edits at each step is illustrated in Table 8, determining the selection
 483 of a I, D, R edit based on its assumed commonsense understanding, which is triggered using a
 484 suitable example.

485 The prompts used by the LVLM to perform the insert and delete edits are provided in Tables 9, 10.
 486 This procedure is needed to ensure commonsense of performed edits. At the same time, it assists the
 487 mask generator of the generative component to define the object that should appear after deleting
 488 another object, effectively handling occlusion, while also masking a suitable area that an existing
 489 object spans in case a new object has to be added in relation to it.

490 C Generative Component

491 In our configuration, object detection operates with a confidence threshold of 0.3, guiding the
 492 inclusion or exclusion of specific object classes via textual prompts. The bounding boxes around
 493 detected objects are expanded by 35 pixels, with a soft boundary applied using a mask blur of 10
 494 pixels. The expansion is required in order for fewer artifacts to emerge from the text prompts, as
 495 further contextual information is added and the areas to be modified are restricted.

496 For inpainting, the process adheres strictly to the provided guidance, with a classifier-free guidance
 497 scale of 10, instructing the model to strongly follow the given prompts. A denoising strength of 1

Table 8: Local edits prompt: defined the operations (I , D , R) that are best to be performed in each step, based on the remaining edits and the image.

I want to remove some objects and add others. I would like you to find the best possible edit for the image, but I want only a single edit.

You can choose from the following options: - Add an object from the "Add" list. In this case please give the answer in the format: ["add", "added_object", "target where the added object will appear in front of"]. Avoid positional description such as "over", "next to", "above" etc. - Remove an object from the "Remove" list. In this case please give the answer in the format: ["remove", "removed_object", "the object that is behind the object when it is removed e.g. wall, floor, background"].

- Replace an object from the "Remove" list with one from the "Add" list. In this case please give the answer in the format: ["replace", "removed_object", "added_object"].

So, you need to decide whether to add, remove, or replace an object.

For example:

Object list: [couch, lamp, window]
Add list: [bed, curtain, blanket]
Remove list: [lamp, couch]

Step: Replace couch with bed.

Another valid step might be:
Step: ["add", "curtain", "window"].

However, the step ["add", "blanket", "couch"] is not a logical step because the couch is on the remove list. If we put the blanket on the couch, we would still have to remove the couch and thus the blanket as well.

Please respond with only a single step and make the most logical edit you can based on the image I have provided.

Object list: objects
Add list: added_objs
Remove list: removed_objs
Step:

Table 9: Prompt defining the addition of objects in the image.

I want to add an object in the image. Please specify what is the object that is target where the added object will appear in front of. Avoid positional description such as "over", "next to", "above" etc. Please respond with a single item, without any additional text. I want to parse this answer automatically, so it is crucial to return only a single object without any explanation, or additional text!

For example:

Add: "painting"
Answer: "wall"
Add: "pillow"
Answer: "bed"
Add: obj
Answer:

498 is used, ensuring the inpainted areas undergo full transformation based on the prompt. The Stable
499 Diffusion v1.5 Inpainting model processes the image for 40 steps, using the a DPM++ 2M SDE
500 sampler, with an automatically chosen scheduler. The pipeline uses a default random seed, ensuring
501 reproducibility with the specification of a fixed seed, while no variation seed is applied, preserving
502 consistency in the output. Additionally, a high-resolution fix is enabled, improving the final image
503 quality through a secondary upscaling pass.

Table 10: Prompt defining the deletion of objects in the image.

I want to remove an object from the image. Please specify what is the object that is behind the object when it is removed e.g. wall, floor, background. Please respond with a single item, without any additional text. I want to parse this answer automatically, so it is crucial to return only a single object without any explanation, or additional text!
For example:
Remove: "painting"
Answer: "wall"
Remove: "pillow"
Answer: "bed"
Remove: obj
Answer:

504 D Qualitative Results

505 In the following Figures, we present some additional qualitative results as occurring from V-CECE
 506 pipeline. Specifically, in Figures 4, 5 we present some successful generations stemming from
 507 DenseNet-suggested edits. DenseNet tends to perform more steps on average in comparison to the
 508 LVLM classifiers (as analyzed in Table 2), which often leads to misgenerations, as the generative
 509 module is unable to handle the complex editing procedure arising as a result of requesting multiple
 510 edits in a row. However, in several cases, DenseNet-driven edits lead to successful counterfactual
 generations, as illustrated below.

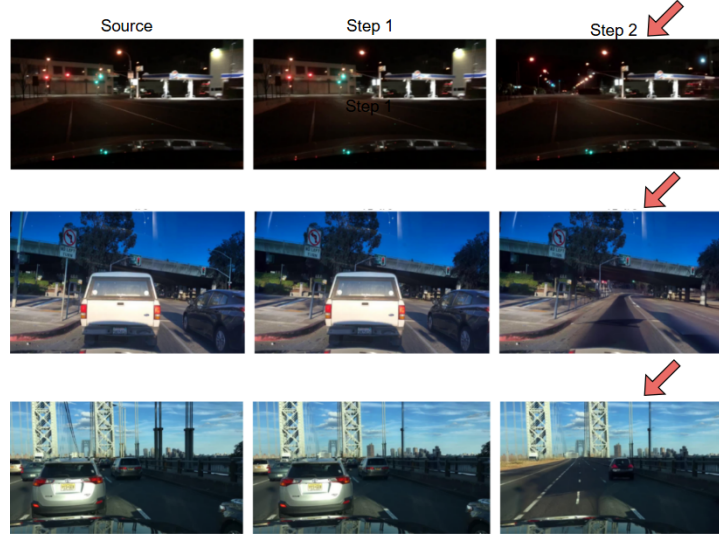


Figure 4: Successful generations after 2 steps of edits for DenseNet classifier. The red arrow denotes the step at which humans perceive label-flipping. In the presented case, DenseNet flips label concurrently with humans and generation terminates.

511



Figure 5: Successful generations after 3 steps of edits for DenseNet classifier. The red arrow denotes the step at which humans perceive label-flipping. In the presented case, DenseNet flips label concurrently with humans and generation terminates.

512 Interestingly, the success of the performed edit is non-trivial, since removing large objects easily
 513 leads to artifacts. Nevertheless, BDD100K images often depict large cars (being close to our point of
 514 view from the driver’s seat), rendering successful edits challenging. This is a reason why prioritizing
 515 the edits in an influential way with respect to the classifier under explanation is crucial.

516 There are some cases where the classifiers need more steps to identify label-flipping, contrary to
 517 humans. Such scenarios are illustrated in Figure 6: classifiers identify label flipping in 3 steps (1st
 518 row) and 2 steps (2nd row) instead of one step that a human perceives as necessary. Therefore, the
 519 classifier instruct the generation module to proceed, leading to irrelevant edits to the class transition
 520 semantics. For example, in the first case, the black car in the same lane as our point of view is
 521 removed in Step 1, allowing the transition from "Stop" to "Move" according to humans. The classifier
 522 however, cannot perceive this change as influential, concluding to a counterfactual image in which
 523 the buildings in the front have been removed, and a black object has been added on the upper right
 524 of the frame. However, these changes are totally irrelevant to the queried driving situation. The
 525 classifier is probably biased towards certain semantics, or even pixel distributions, therefore being
 526 fooled under such transformation, instead of flipping during the removal of the black car in Step 1.
 527 In the second case, the white car in the front is removed at Step 1, correctly marked by humans as a
 528 "Move" situation. The classifier instructs further generation, resulting in the replacement of the tree
 529 with a street light in the front. Nevertheless, this semantic edit is not associated with whether one has
 530 to brake or move, deeming this operation as an extraneous edit, wrongly imposed by limited semantic
 531 comprehension of the classifier. In the last case, human and classifier perception of label-flipping
 532 agree, since the removal of the car in the front suggests transiting to the "Move" class. However, we
 533 observe a visual artifact in place of the big car. This example denotes the limitations of the generation
 534 module employed in our experimentation, suggesting that even if a single step is performed towards
 535 counterfactual generation, it is not guaranteed that the resulting image will be of good quality. Once
 536 again, removing large objects is a tough endeavor itself for visual editors, and it is rather unpredictable
 537 whether this operation will be performed without any detectable artifact.



Figure 6: Interesting cases of sub-optimal counterfactual generations. The red arrow denotes the step at which humans perceive label-flipping. In the first two cases, classifiers flip label later than humans; therefore, generation terminates later than necessary. In the last case, humans and classifier perception align, but generation is not devoid of artifacts.

E Human survey

Our human survey on BDD100K generated counterfactual images was filled by 31 participants. We gathered no personal information about these evaluators. We used the Label Studio platform for evaluation, allowing us to demonstrate image sequences, along with the required descriptions and questions. Specifically, the participants were provided with a source image and a sequence of numbered generation steps, as occurring from our experiments (we incorporated all classifiers and all ordering techniques). They were then asked to respond to the following:


- The step at which they believe label flip is happening, given that the source class is always "Stop". If label flip did not happen at all in this specific image sequence, they can reply with "None of the above".
- The visual correctness of the image, given the options "Yes" (if the image is visually correct, meaning that it is absent of severe visual artifacts) and "No".

An example of the questionnaire they were asked to fill is presented in Figure 7.


You will evaluate images from driving scenes where the cars should either be in 'Move' or 'Stop' mode. Imagine that you are deciding whether to move or stop the car as you review a sequence of images. The class of the source image is always 'Stop'.

1) In the first task, select the step where you transition from 'Stop' to 'Move', if this change occurs.
2) The second question assesses if the final image in the sequence is visually correct, meaning it should not have severe visual artifacts that indicate the image is generated (ignore other characteristics such as low resolution or low-light conditions). If the image is visually correct, answer 'Yes'; otherwise, answer 'No'.

source.jpg



step_1.jpg



At which step do you think the classification label changes? For example, if steps 1 to 4 belong to the class 'Stop' and step 5 belongs to the class 'Move,' select 'Step 5,' even if the class changes again in subsequent steps. If all the images belong to the class 'Stop,' select 'None of the above.'

☐ Step 1¹¹
☐ Step 2²²
☐ Step 3³³
☐ Step 4⁴⁴
☐ Step 5⁵⁵
☐ Step 6⁶⁶
☐ Step 7⁷⁷
☐ Step 8⁸⁸
☐ Step 9⁹⁹
☐ Step 10¹⁰¹
☒ None of the above¹¹

Is the final image in the sequence visually correct, meaning it does not have severe visual artifacts that indicate it is generated (ignore other factors such as low resolution or low-light conditions)? If the image is visually correct, answer 'Yes'; otherwise, answer 'No.'

☐ Yes¹¹
☒ No¹¹

Figure 7: Panel of a human annotation instance in Label Studio.

Consequently, we delve into the human evaluation results, since they are crucial in unveiling the explanatory gap between humans and classifiers via V-CECE explanations. Therefore, we analyze human responses regarding visual correctness (Figures 8, 9, 10) and steps required (Figures 11, 12, 13) until label flip for each ordering method, as well as average values for all methods.

Commencing with DenseNet classifier in Figure 8, its average correctness lies around 60% based on human perception of visual quality. Regarding the ordering techniques for edits, local edits, instructed by Claude 3.5 sonnet on the proposed edit set $|E|$, as occurring from the explanation component, arises as the most successful strategy with 64.58% successful counterfactual generations. The most 'greedy' strategies (with respect to label flipping) that consult global edits score lower, with 57.89% for global and 56.67% for local-global edits.

The local edits are proven as the most successful also in the case of Claude 3 Haiku human results in Figure 9, achieving a 73.47% on visual correctness. On average, Claude 3 Haiku achieves 69.62% correctness indicating a medium agreement with human perception in semantic comprehension for classification.

The patterns changes when Claude 3.5 Sonnet is leveraged as the classifier, where local edits results in only 73.97% correctness, scoring lower than the average of 78.3% on all orderings. Local-global edits lead to 87.88% correctness, the highest percentage overall, suggesting that leveraging model biases in conjunction to LVLm-driven ordering is the best practice for this classifier. Global edits achieve 77.42% correctness, indicating that a 'greedy' edit selection choice is effective, though sub-optimal without proper ordering.

The average number of steps needed for label flip is an informative indicator for the classifier's semantic level as demonstrated on the human survey findings (Table 2). Single-step edits are the most

Visual Correctness on DenseNet

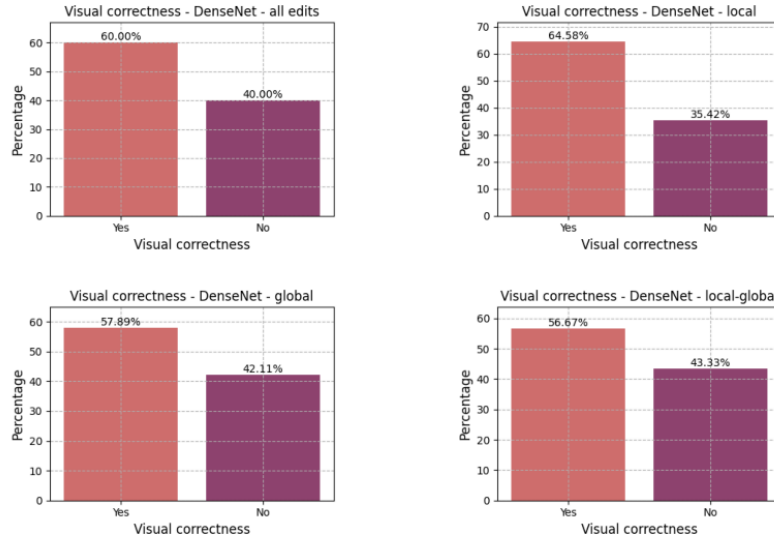


Figure 8: Human evaluation results regarding visual correctness with edits driven from DenseNet classifier.

Visual Correctness on Claude 3 Haiku

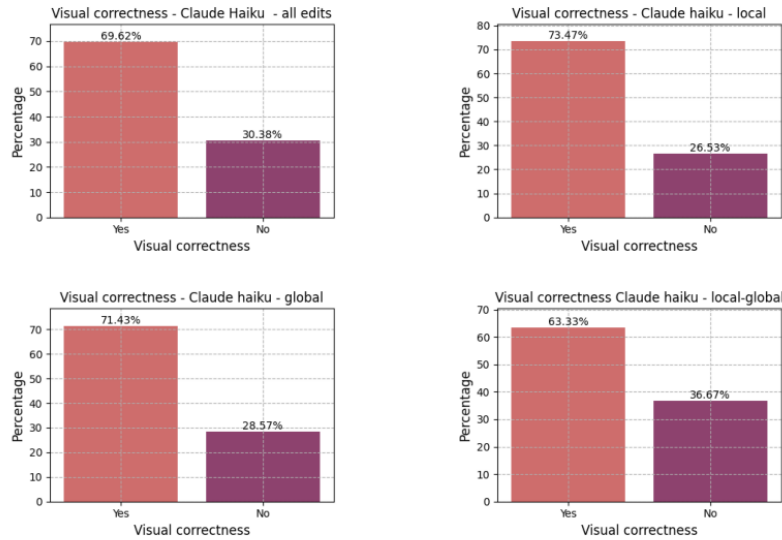


Figure 9: Human evaluation results regarding visual correctness with edits driven from Claude 3 Haiku classifier.

573 prevalent on average for DenseNet. Interestingly, when local edits are employed, the label-flipping
574 procedure needs two steps as the most frequent step frequency. At the same time, local edits are
575 associated with the best-quality generations for DenseNet, suggesting that despite often needing
576 two steps, the finally generated images are as good as possible, in comparison with other ordering
577 strategies. Furthermore, local and local-global strategies for DenseNet never require more than 5
578 steps for label flipping, contrary to global edits, which presents few cases of 6 and 7 edits. This
579 finding verifies the effectiveness of Claude 3.5 Sonnet as an edit ordering module, which assists in
580 driving counterfactual generations in fewer steps, thanks to its contextual and spatial understanding.

Visual Correctness on Claude 3.5 Sonnet

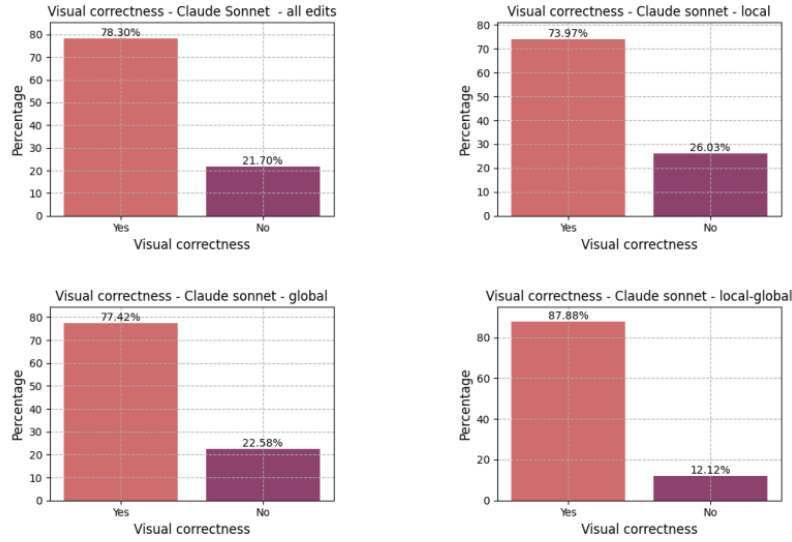


Figure 10: Human evaluation results regarding visual correctness with edits driven from Claude 3.5 Sonnet classifier

Regarding Claude 3 Haiku (Figure 12), single-step generations are the most frequent scenario. The behavior of this classifier is more predictable, demonstrating often 2 or 3-step generations, but with a striking difference in comparison to the single-step ones. In very few cases, 7 or 8 steps are needed, associated with local and local-global orderings, while for global edits, the steps are at most 5 in few instances. Global edits impose a more aggressive editing strategy towards label flipping, as indicated in Figure 12, but this does not mean these edits are reasonable with respect to the source image semantics, a finding that is cross-verified by the lower image correctness reported previously in Figure 9.

Finally, Claude 3.5 Sonnet presents an outstanding dominance of single-step generations as the most frequent case, as exhibited in Figure 13. Very few cases require more than one step to change classification label and are primarily associated with the local edits strategy (and secondly with the local-global ordering). This verifies that the edits suggested by Claude 3.5 Sonnet in each generation step are suboptimal, agreeing with the visual correctness findings of Figure 10. On the contrary, all generations driven by global edits need only 1 step until label flipping, highlighting this ordering strategy as the most successful one for Claude 3.5 Sonnet classifier, both in terms of editing steps and visual correctness.

Number of steps for label flip on DenseNet

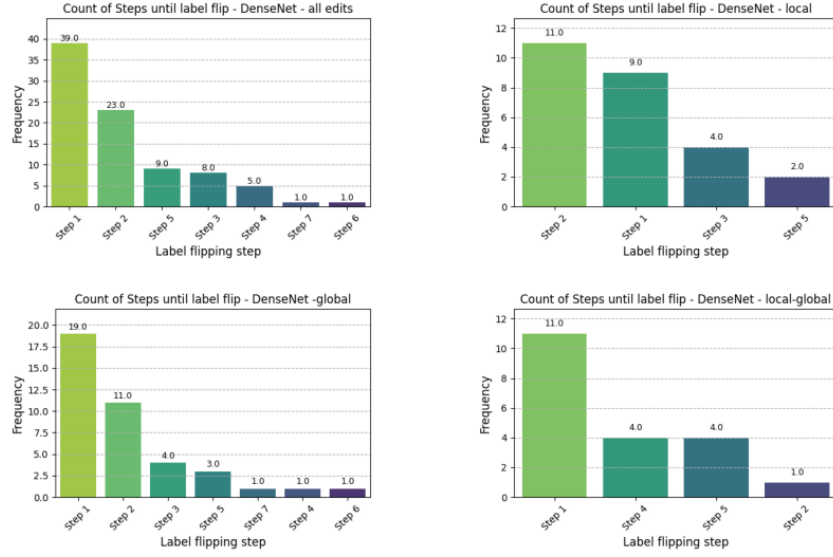


Figure 11: Number of steps until label flip distribution for DenseNet-driven edits.

Number of steps for label flip on Claude 3 Haiku

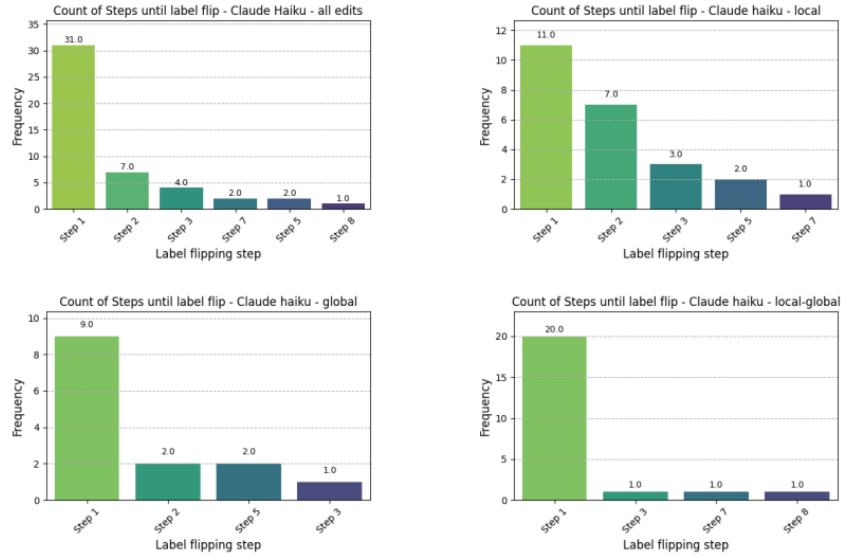


Figure 12: Number of steps until label flip distribution for Claude 3 Haiku-driven edits.

Number of steps for label flip on Claude 3.5 Sonnet

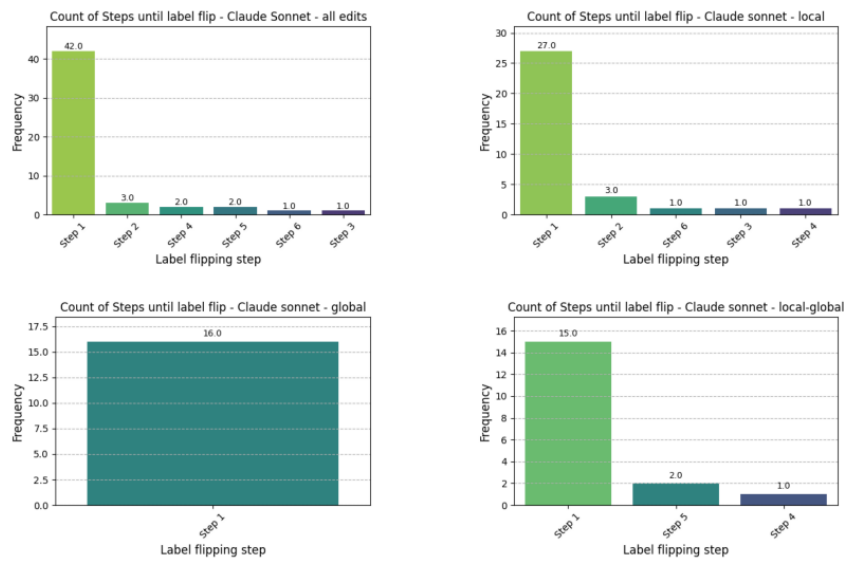


Figure 13: Number of steps until label flip distribution for Claude 3.5 Sonnet-driven edits.