# EmoEdit: Evoking Emotions through Image Manipulation (Supplementary Material)

Jingyuan Yang[1], Jiawei Feng[1], Weibin Luo[1], Dani Lischinski[2], Daniel Cohen-Or[3], Hui Huang[1]*
[1]Shenzhen University [2]The Hebrew University of Jerusalem [3]Tel Aviv University
{jingyuanyang.jyy, fengjiawei0909, waibunlok, danix3d, cohenor, hhzhiyan}@gmail.com

## 1. Construction Details

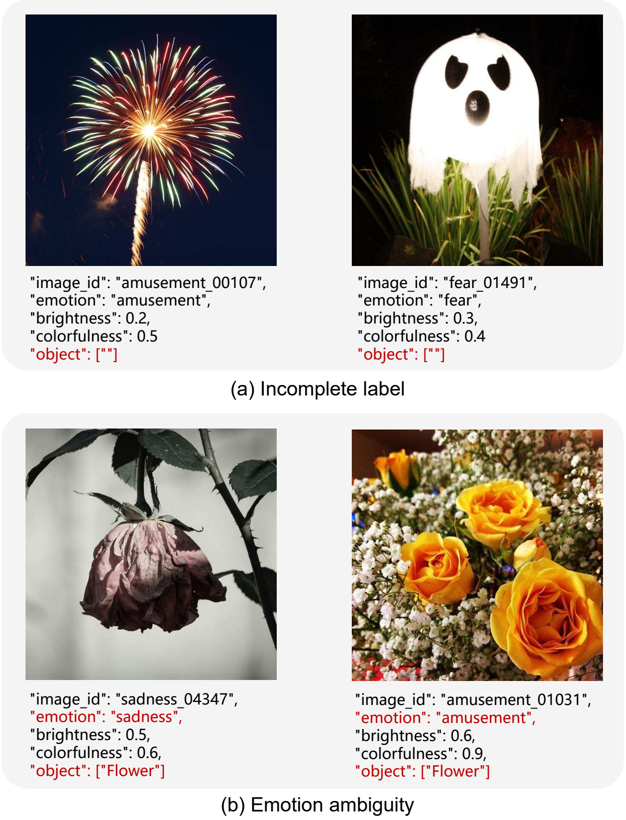
In this section, we present a detailed overview of the network construction, encompassing emotion attribution, data construction and the design of Emotion adapter.

### 1.1. Emotion Attribution

EmoSet is one of the largest visual emotion datasets, featuring rich attribute labels across diverse categories such as color, lighting, objects, scenes, human actions, and facial expressions. However, the attribute labels in EmoSet are limited in accurately capturing the diverse range of emotional expressions. As shown in Fig. 1, some important emotional labels, such as "firework" and "ghost", are missing due to the limited categories in the pre-trained attribute models. Additionally, EmoSet labels each image with a single word, which significantly limits the accuracy of emotional expressions. For instance, wilted flower and blooming flower are both labeled with "flower", leading to the emotional ambiguity, *i.e.*, *sadness* and *amusement*.

Consequently, we conduct clustering on EmoSet to identify the common visual cues for each emotion. As shown in Algorithm 1, we first initialize each cluster with an image, followed by iteratively merging two clusters with the highest similarities until all inter-cluster similarities are below 0.89. However, due to its unsupervised nature, the clustering results are not consistently perfect. Consequently, we implement several post-processing steps to eliminate emotion-agnostic clusters. Clusters containing fewer than five images are deemed unimportant for the associated emotion category. Additionally, clusters with images exhibiting excessive similarity at the pixel level are considered inadequate as generalized semantic factors. To enhance the efficacy of emotion editing, we exclude clusters that fail to evoke emotion. Detailed steps of the filtering process are outlined in Algorithm 1.

We chose the GPT-4V model as our VLM due to its advanced capabilities in multimodal contextual understanding. The given prompt is presented in Table 1. For each remaining cluster after the filtering process, we provide GPT-



(a) Incomplete label

(b) Emotion ambiguity

Figure 1. EmoSet has limitations in its attribute labels. Some important attributes are incomplete, and the use of single-word labels can lead to emotional ambiguity.

Table 1. Prompts for GPT-4V summarizing clusters in EmoSet.

| Stage | Prompt |
|---|---|
| Factor summary | Generate a concise description of the commonality of an image collection, focusing on either objects or actions using sentence of 3 words. Identify the main element involved, either the object or the person engaged in the action. This clear, focused format improves compatibility with image generation models by specifying distinct elements.<br>3 words: <description><br>Main element: <object or person involved in action> |

*Corresponding author

4V with the five images closest to the centroid, asking it to generate a concise content-color summary of no more than three words each. The detailed prompt used to interact with GPT-4V is shown in Table 1.

Consequently, we construct eight emotion factor trees, where each represents an emotion category, as shown in Fig. 11 and Fig. 12. Specifically, four positive emotions (*amusement*, *awe*, *contentment*, and *excitement*) are depicted in Fig.11, while four negative emotions (*anger*, *disgust*, *fear*, and *sadness*) are depicted in Fig.12. Most emotion factor trees include four types of factors: objects, scenes, actions, and facial expressions, illustrating the content diversity of EmoSet. For example, factors such as "Pink blooming roses", "Festive holiday decorations" and "Kids birthday party" can evoke people's emotion of *amusement*. Conversely, "Autumn leaves scattered", "Cemetery tombstones outdoors" and "Abandoned dilapidated interiors" may evoke feelings of *sadness*.

## 1.2. Data Construction

There are several unexpected issues encountered during the generation of data pairs, such as high semantic similarity and high content abstractness. High semantic similarity refers to cases where certain emotion factors within the same emotion factor tree are highly alike and can be consolidated into a single factor, *e.g.*, "Smiling children" and "Laughing babies". High content abstractness arises from the knowledge gap between GPT-4V and Stable Diffusion. Specifically, GPT-4V tends to generate complex, high-level language that Stable Diffusion struggles to interpret effectively, *e.g.*, "Lying in grass" and "Couples sharing affections".

**CLIP Metrics** We introduce CLIP metrics, namely CLIP image similarity (CLIP-I) and CLIP text similarity (CLIP-T), to assess the similarity between the input image, text prompt, and edited image. Specifically, CLIP-I calculates the cosine similarity between the feature representations of two images, while CLIP-T evaluates the alignment between text and image features. The input image $x_{input}$ and the edited image $x_{edit}$ are first encoded by the CLIP image encoder $\mathcal{E}_{img}(\cdot)$ and further calculated as

$$CLIP-I = \frac{\mathcal{E}_{img}(x_{input}) \cdot \mathcal{E}_{img}(x_{edit})}{\|\mathcal{E}_{img}(x_{input})\|_2 \|\mathcal{E}_{img}(x_{edit})\|_2}, \tag{1}$$

$$CLIP-T = \frac{\mathcal{E}_{img}(x_{edit}) \cdot \mathcal{E}_{txt}(t_{ins})}{\|\mathcal{E}_{img}(x_{edit})\|_2 \|\mathcal{E}_{txt}(t_{ins})\|_2}, \tag{2}$$

where $\mathcal{E}_{txt}(\cdot)$ represents the CLIP text encoder, and $t_{ins}$ denotes the content instruction.

**ALGORITHM 1:** Emotion Attribution Algorithm

```
Input: $M$ images: $D = \{x_1, ..., x_M\}$
Output: $K$ clusters: $C = \{C_1, ..., C_K\}$; $N$ factors :
        $C' = \{C'_1, ..., C'_N\}$
Step 1: Semantic Clustering
Extract CLIP feature for each image:
  $f_m = CLIP_{visual}(x_m)$;
Initialize $M$ clusters each with an image:
  $C_m = \{f_m\}$;
Calculate similarity matrix $S(i,j) = sim(C_i, C_j)$;
while $\max(S(i,j)) \geq 0.89$ do
    Merge clusters in $a, b = arg\max(S(i,j))$ as a
      new cluster $c$;
    Calculate the centroid of cluster $c$;
    Update similarity matrix $S$;
end
return K clusters: $C = \{C_1, ..., C_K\}$
Step 2: Emotion Filtering
Initialize emotion factor index: $n = 0$;
for $k = 1$ to $K$ do
    if $|C_k| \geq 5$ then
        Calculate averaged similarity:
          $s_k = \frac{1}{|C_k|}\sum_{i,j=1}^{|C_k|}(sim(f_k^i, f_k^j))$;
        if $s_k \leq 0.89$ then
            Calculate averaged emotion score:
              $e_k = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|}(F_{emo}(f_k^i))$;
            $F_{emo}$ is a pre-trained emotion classifier;
            if $e_k \geq 0.3$ then
                cluster k survives the filtering
                  process: $C'_n = C_k$;
            end
        end
    end
end
return N factors: $C' = \{C'_1, ..., C'_N\}$
```

**Aesthetic Score** Image quality should prioritize human preferences over traditional image reconstruction metrics. In affective image manipulation (AIM), special emphasis must be placed on the image's aesthetic appeal to effectively evoke emotions. To address this, we further introduce an aesthetic score to filter out undesirable images, such as those with distorted content or poor composition. The aesthetic model is derived from a GitHub project[*] released by LAION-AI, utilizing CLIP features for classification.

**Emotion Score** Since our task is AIM, we train an emotion classifier in the CLIP space using EmoSet, achieving an overall accuracy of 83%. The high accuracy indicates

[*] https://github.com/LAION-AI/aesthetic-predictor

that the pre-trained emotion classifier can effectively distinguish between different emotions. To be specific, the classifier consists of a frozen CLIP image encoder followed by a trainable fully connected layer. We leverage this classifier to compute emotion score:

$$S_{emo}(x, y_{emo}) = y_{emo} \cdot (p(\varphi(x))), \tag{3}$$

$$p(q_i) = \frac{\exp(q_i)}{\sum\limits_{j=1}^{C} \exp(q_j)}, \tag{4}$$

where $y_{emo}$ denotes the one-hot emotion label, $\varphi(\cdot)$ represents the emotion classifier, $q_i$ refers to the $i$-th component of the vector, $C$ indicates the total number of emotion categories, $x$ depicts the input image.

After the construction process, EmoEditSet is built as shown in Fig. 5. It contains eight emotion categories, each with various semantic variations. For example, the emotion category *awe* includes content instructions such as "Fountain rainbow", "Snow-covered volcano", "Northern lights display", and "Colorful hot-air balloons", all of which evoke the same emotion.

### 1.3. Emotion Adapter

To explore the correlation between the input image and the target emotion, we design the Emotion adapter based on the Q-Former. Specifically, the input image and target emotion are encoded separately using the CLIP image and text encoders, respectively, with both represented by CLIP embeddings. The target emotion feature is then concatenated with the learnable query embedding, forming a (77, 768) dimensional input for the Q-Former, while the input image feature serves as the input for cross-attention layers. Finally, we incorporate a LayerNorm layer at the Q-Former's output to enhance feature normalization and stability.

In the overview of EmoEdit, there are two distinct network designs: General and Emotional. The General component follows the conventional pipeline typically used in text-to-image models, while the Emotional component illustrates how EmoEdit integrates emotional knowledge into the editing process.

## 2. Experimental Details

In this section, we present the experimental details, including the formulas for the specialized emotion metrics (Emo-A, Emo-S), the prompts employed in the comparison methods, and the specifics of the ablation study and user study.

### 2.1. Implementation Details

Our experiments are conducted using PyTorch on eight Nvidia RTX 4090 GPUs, each with 24GB of memory. We utilize the pre-trained InstructPix2Pix model[†] for data construction and EmoEdit, while the CLIP ViT-L/14 model[‡] is used for the emotion classifier and the Emotion Adapter. The Emotion Adapter is trained for 30,000 steps over a period of 20 hours. Our model is trained at a resolution of 256 × 256 with a total batch size of 256. The learning rate is set to $10^{-4}$ without any warm-up. For inference, the results presented in this paper are generated at a resolution of 512 with 30 denoising steps.

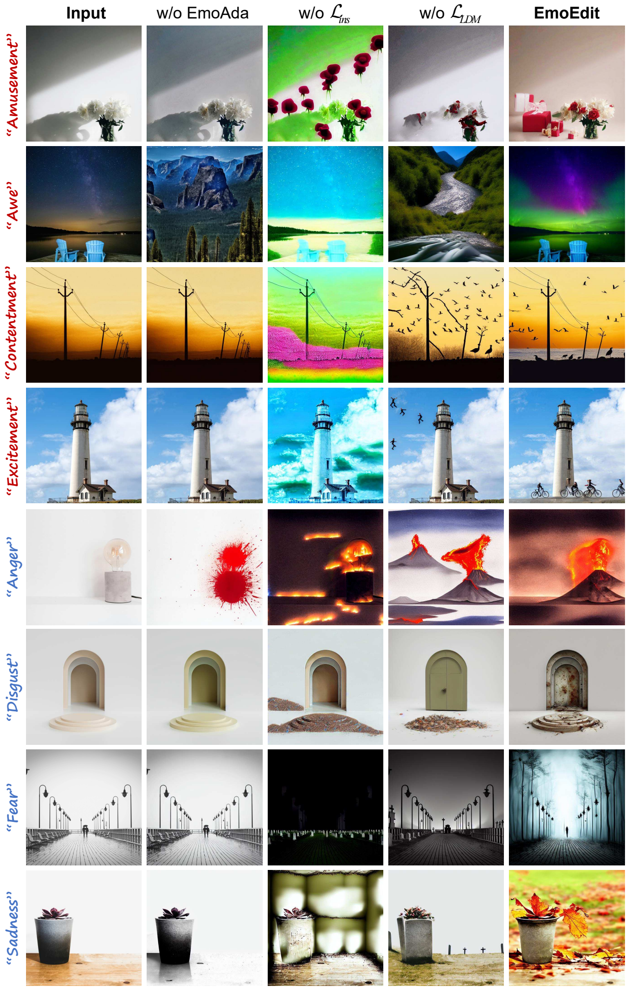


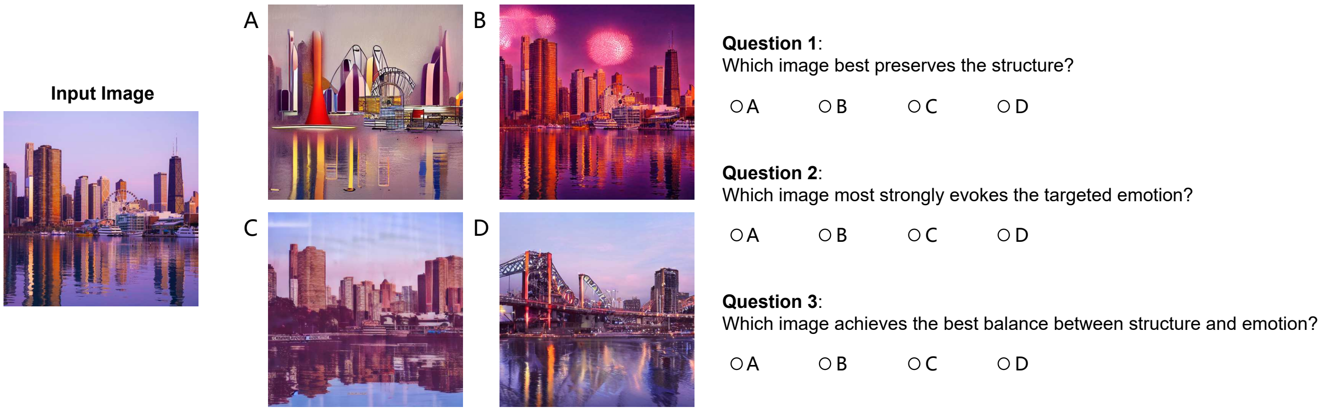
Figure 2. Additional results on ablation study, where eight emotions are involved.

### 2.2. Evaluation Metrics

Among all the evaluation metrics, Emo-S is first introduced by EmoEdit, and a detailed explanation is provided for better understanding.

[†] https://huggingface.co/timbrooks/instruct-pix2pix

[‡] https://huggingface.co/openai/clip-vit-large-patch14



**Question 1**:
Which image best preserves the structure?

○A ○B ○C ○D

**Question 2**:
Which image most strongly evokes the targeted emotion?

○A ○B ○C ○D

**Question 3**:
Which image achieves the best balance between structure and emotion?

○A ○B ○C ○D

Figure 3. The interface of user study. Given an input image and four edited images, users are asked three questions on emotion fidelity and structure integrity.

**Emo-S** Unlike generation tasks, AIM is more challenging because it aims to evoke different emotions through only minor modifications to the original input. This goal is inherently contradictory: to significantly evoke emotions while remaining faithful to the original structure. To address this, we propose a new metric for AIM called the Emotion Incremental Score (Emo-S) to evaluate the increase in the desired emotion score. Specifically, Emo-S calculates the difference in emotion scores between the input image and the edited image, focusing on a specific emotion type:

$$Emo-S = S_{emo}(x_{edit}, y_{emo}) - S_{emo}(x_{input}, y_{emo}), \tag{5}$$

while $x_{input}$ represents the input image, $x_{edit}$ depicts the edited image.

## 2.3. Comparisons

As the first to explore content editing in AIM, our comparison methods are related but do not directly address the same problem. Additionally, the compared methods differ in their prompt forms and emotional settings, and we categorize them into three input types: description-based, prompt-based, and emotion-based. For description-based approaches, *i.e.*, SDEdit, PnP, BlipDiff, we use the prompt "An image" to represent the input image and the target image prompt with an emotional trigger, *e.g.*, "An image that evokes the emotion of [target emotion]". For instruction-based approaches, *i.e.*, InsDiff, ControlNet, we give a direct instruction to these methods, *i.e.*, "Add elements that evokes the emotion of [target emotion]". Notably, ControlNet is trained on IP2P dataset, provided by the official team in huggingface[§]. Emotion-based approaches, *i.e.*, CLVA, AIF, require emotional descriptions as input, so we randomly select one from their affective description set based on the target emotion.

## 2.4. User Study

We recruit 41 healthy Asian volunteers, aged between 22 and 56. Fig. 3 illustrates the interface used in the user study. Participants are required to answer three questions on emotion fidelity and structure integrity, with the positions of the four images randomly shuffled to ensure a fair comparison.

## 2.5. Applications

We show the potential of Emotion adapter to enhance emotion awareness of various diffusion-based models, encompassing editing tasks and generation tasks.

For text-to-image editing models, we replace their original input text condition with the output of the Emotion adapter. Most comparison methods are included, except for CLVA, AIF, and BlipDiff, as these either are not diffusion-based or require manipulation of text prompts.

The setup for the generation task differs significantly from that of the editing task. In the generation task, since no input image is provided, we randomly select an image from EmoSet and obtain the required emotion embedding through the Emotion Adapter. Given the involvement of multiple conditions, namely emotion and style, we introduce Composable Diffusion to achieve the desired outcome.

# 3. Additional Results

In this section, we present various results on EmoEdit, ranging from eight-direction editing (Fig. 4), diverse semantics (Fig. 6), qualitative comparisons (Fig. 7), ablation study (Fig. 2), emotion-enhanced editing model (Fig. 8, Fig. 9) and emotion-aware stylized image generation (Fig. 10).

[§] https://huggingface.co/lllyasviel/control_v11e_sd15_ip2p

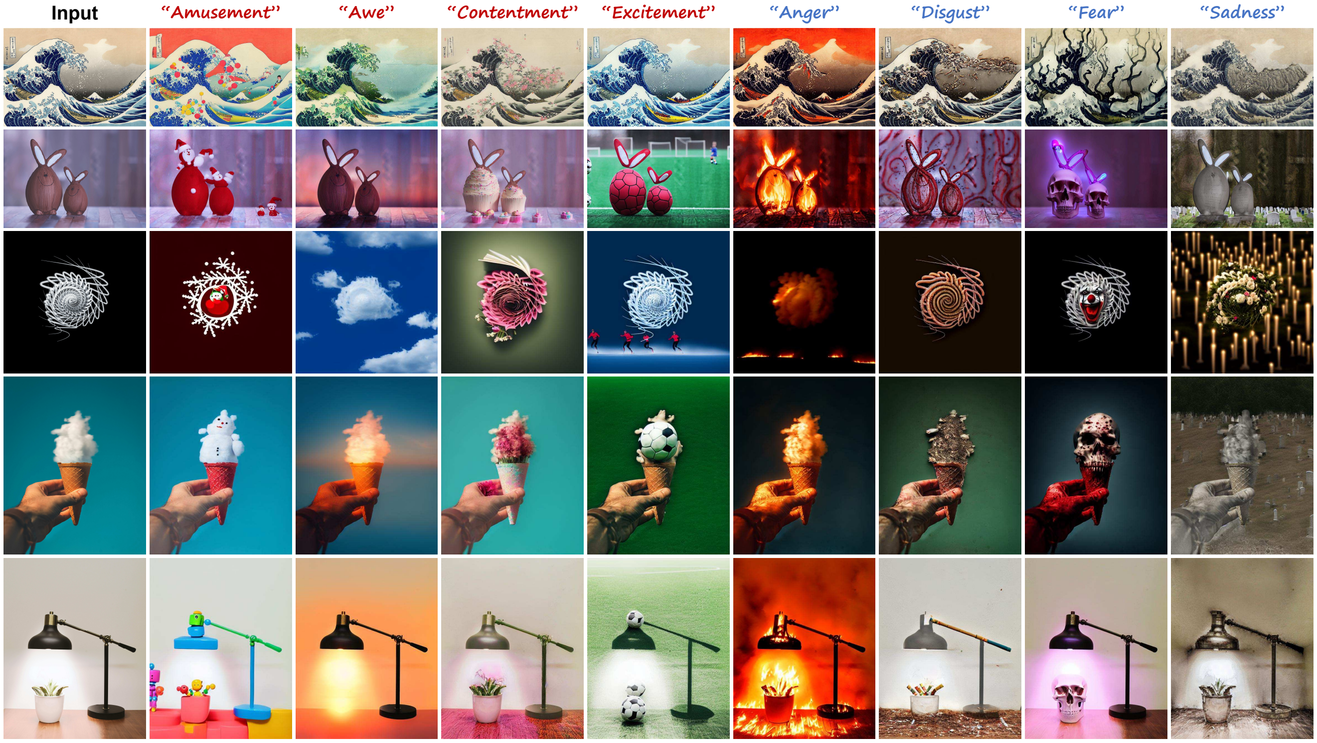


Figure 4. Eight-direction emotion editing. Given one input image, EmoEdit is capable to modify it to eight different emotion directions.

### 3.1. Eight-direction Editing

EmoEdit can modify any user-provided image to express different emotional directions, as illustrated in Fig. 4. In the second row, for instance, an image of two rabbits is transformed into a snowman to evoke *amusement*, into flames to evoke *anger*, into skulls to evoke *fear*, and into a cupcake to evoke *contentment*, demonstrating how distinct semantic elements can trigger different emotions. Conversely, in the second column, various elements, such as balloons, Santa, snow, a toy robot, and a snowman, are used to evoke *amusement*. These results highlight that: (1) EmoEdit can edit a single input image in eight emotion directions; (2) EmoEdit can generate diverse semantic variations within a specific emotion direction.

### 3.2. Diverse Semantics

Fig. 6 showcases EmoEdit's editing results across eight distinct emotions. For *awe*, the semantics include elements like auroras, churches, Indian castles, and sunsets. Besides, *contentment* is represented by elements such as coffee, swimming pools, flowers, and cakes. The *fear* category features foggy forests, clowns, skulls, and pumpkins, while *sadness* is conveyed through graveyards, candles, autumn leaves, and dilapidated houses. These results demonstrate that modifications for each emotion are not over-fitted to a single content instruction. Instead, each emotion type is represented by a diverse range of semantic elements, enhancing both the effectiveness and expressiveness of AIM.

### 3.3. Other Results

We also expand the results on qualitative comparisons, ablation study, emotion-enhanced editing model and emotion-aware stylized images generation to eight emotion categories. These results support the conclusions drawn in the main paper: (1) Most comparison methods lack emotional knowledge and are prone to distortion (Fig. 7); (2) Both instruction loss and diffusion loss are crucial for achieving clear semantics and structure preservation (Fig. 2); (3) The Emotion Adapter can be plugged into existing diffusion-based editing models to enhance emotion-awareness (Fig. 8, Fig. 9); (4) The Emotion Adapter can also be incorporated into existing diffusion-based generation models to evoke emotions while preserving the specified style (Fig. 10). Furthermore, we demonstrate EmoEdit's potential across eight emotional directions in each experiment. These results, compared with the baseline outcomes in the main paper, further validate EmoEdit's effectiveness and robustness.

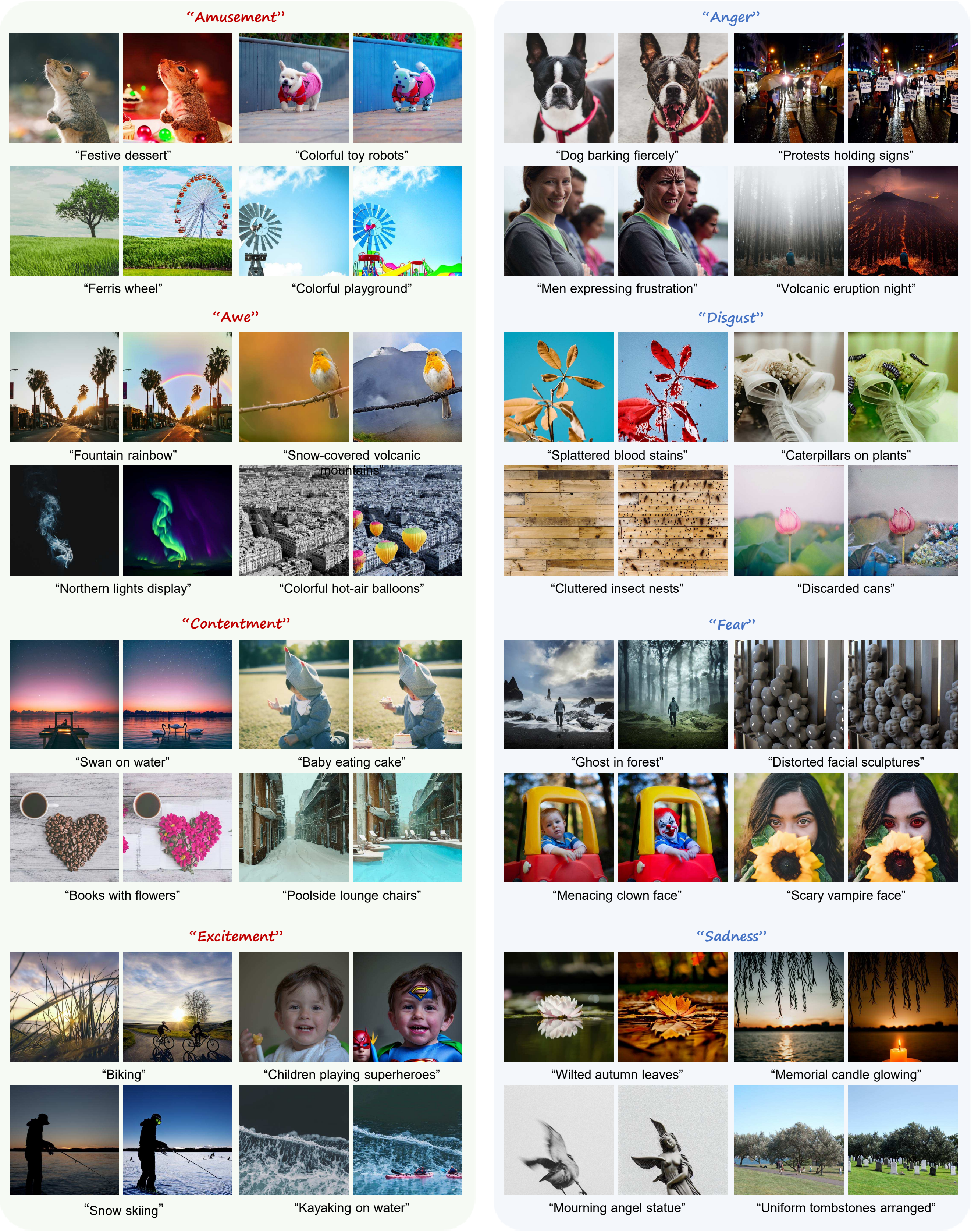


Figure 5. Image data pairs in EmoEditSet, where each labeled with an emotion direction (top) and a content instruction (bottom).

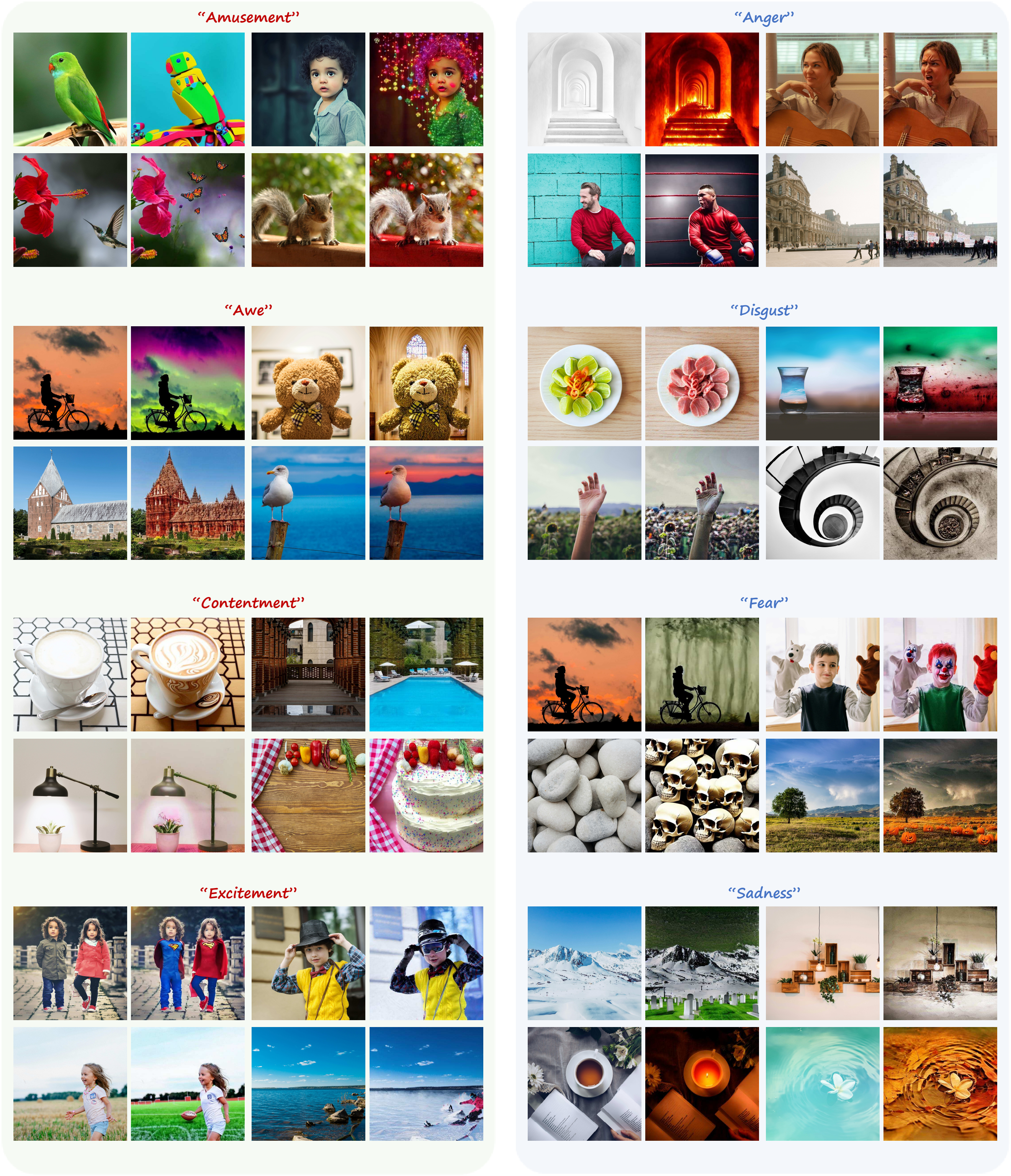


Figure 6. Editing results of EmoEdit, where several semantic variations are presented within one specific emotion.

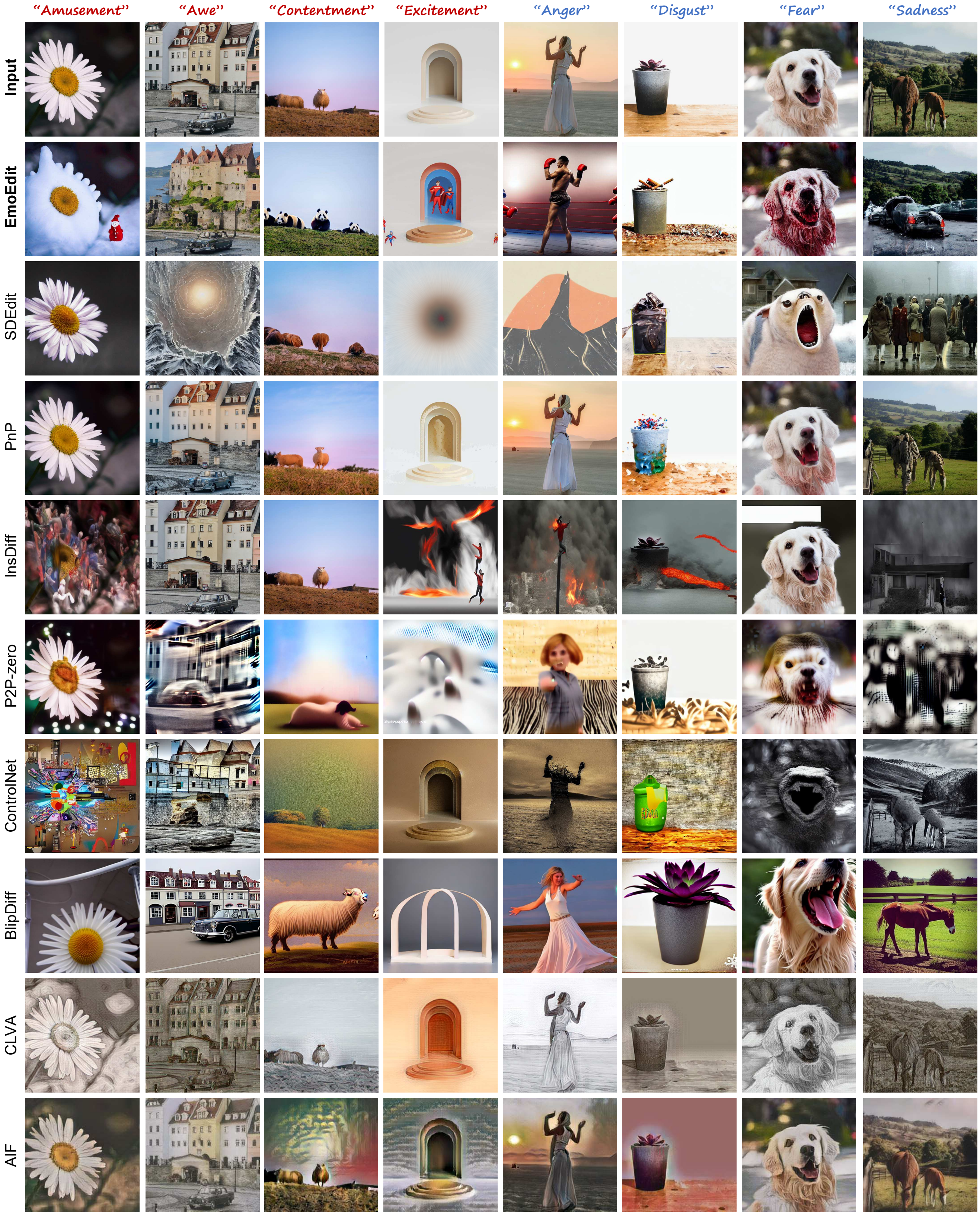
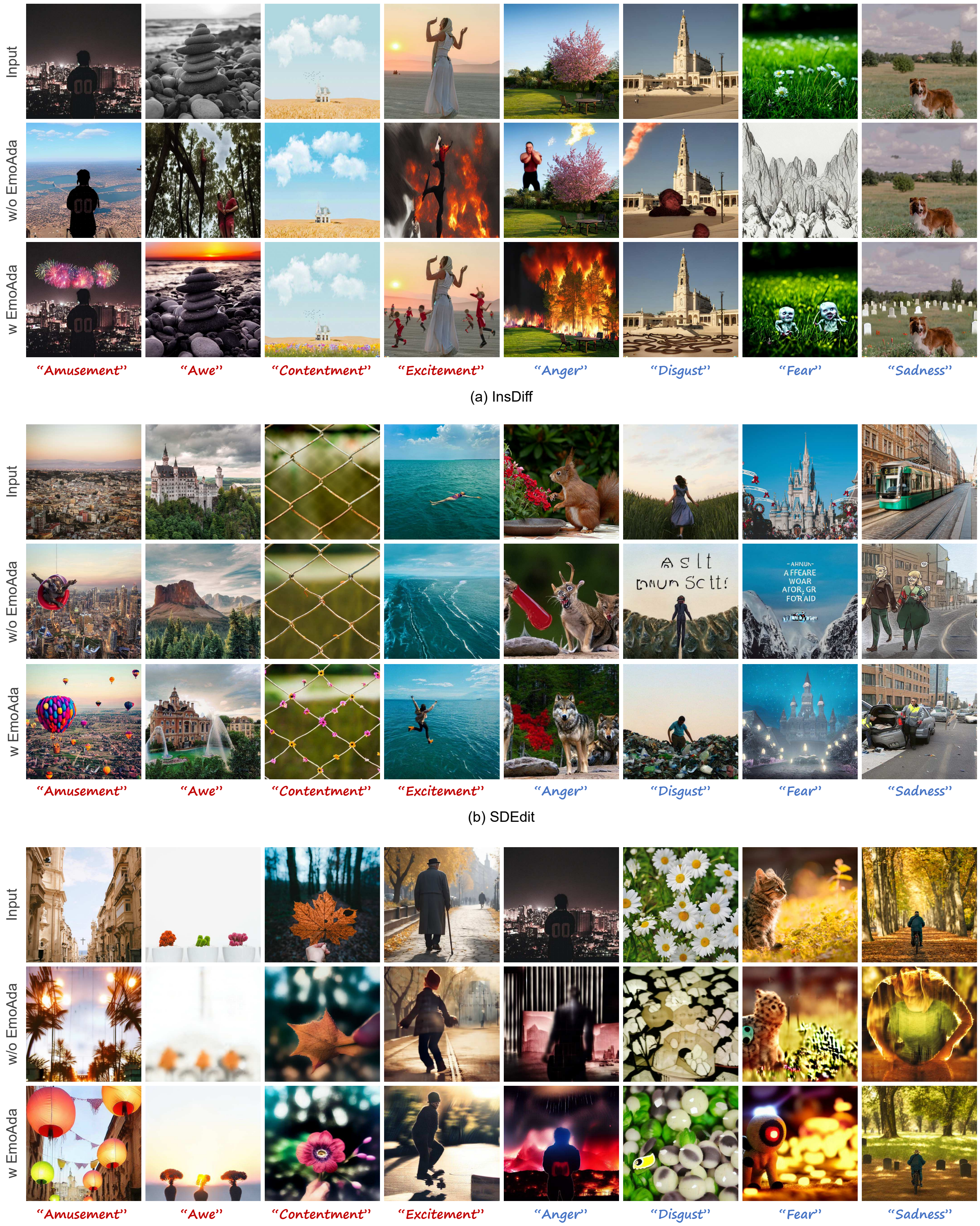


Figure 7. Additional results on qualitative comparisons, where eight emotions are involved.

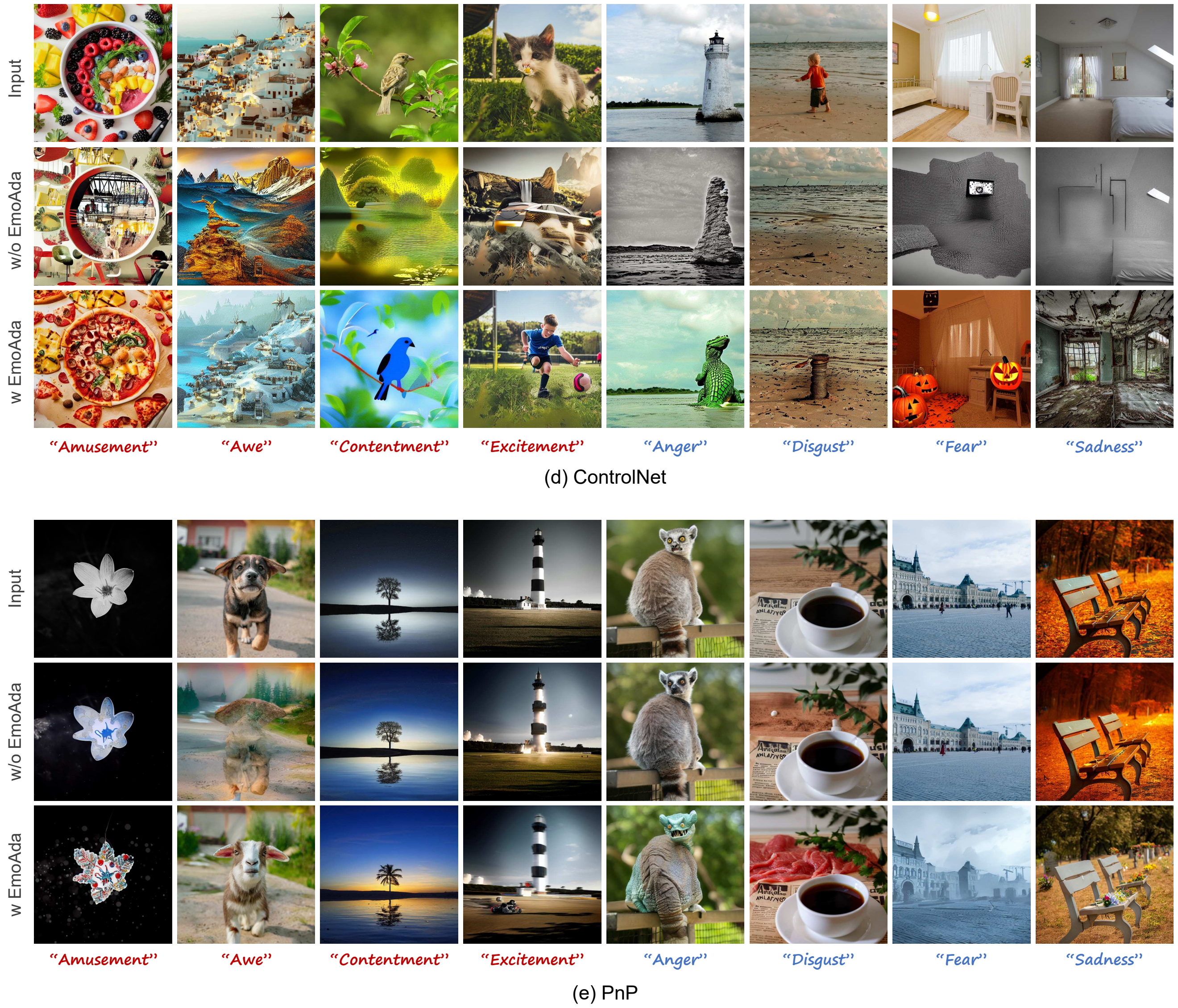


(a) InsDiff

(b) SDEdit

(c) P2P-zero

Figure 8. Additional results on emotion-enhanced editing models, where eight emotions are involved.



(d) ControlNet

(e) PnP

Figure 9. Additional results on emotion-enhanced editing models, where eight emotions are involved.

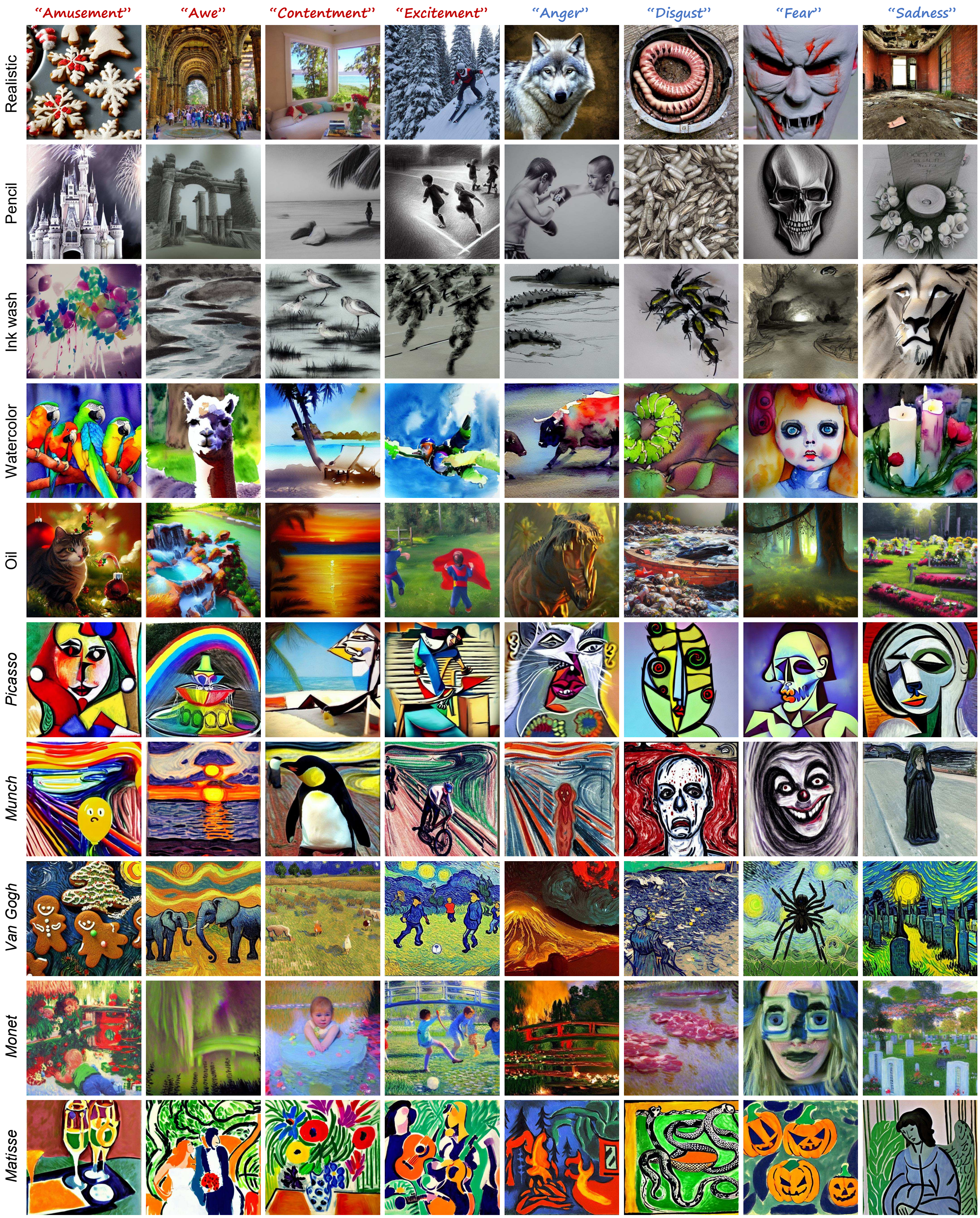
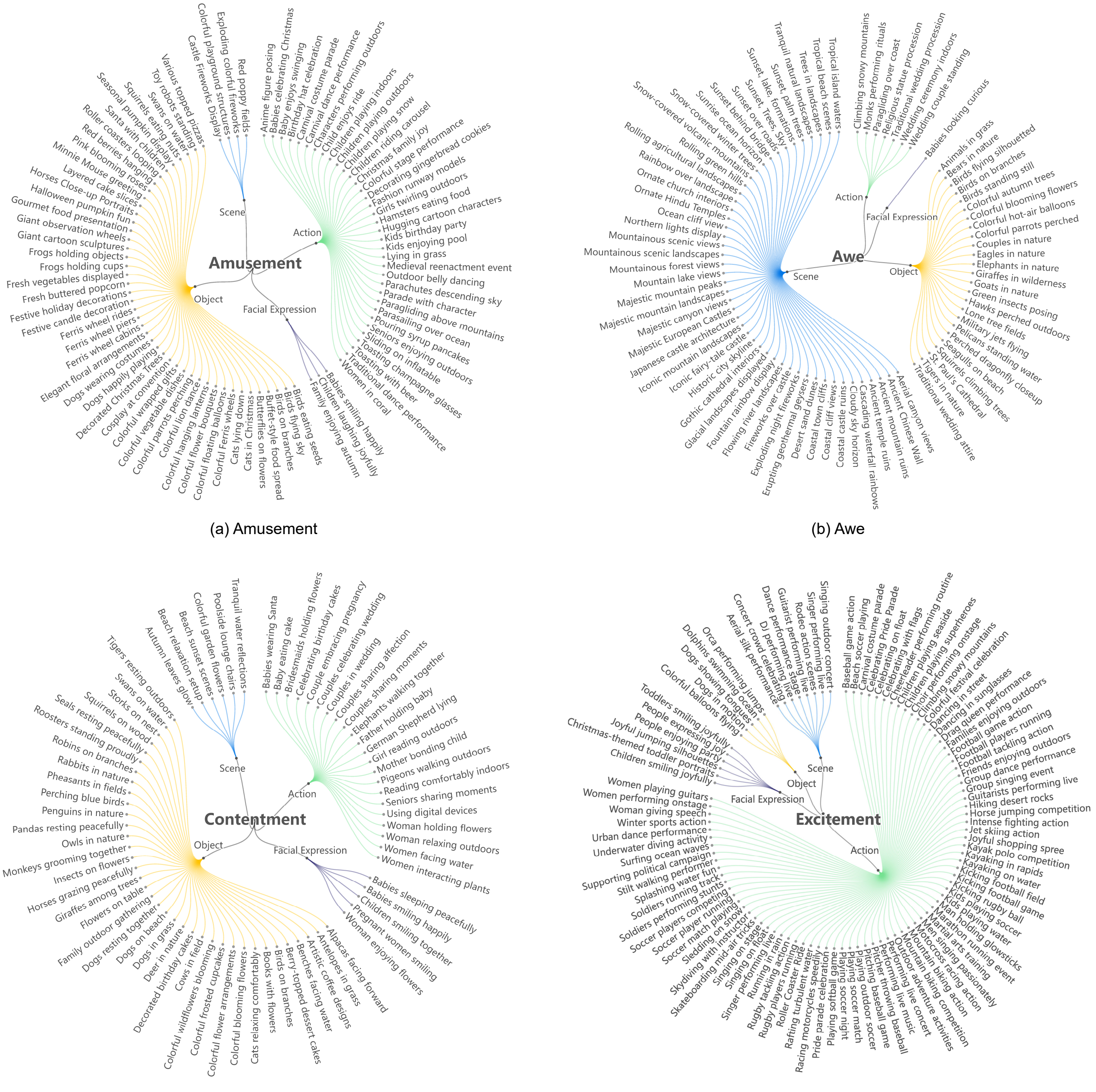


Figure 10. Additional results on emotion-aware stylized image generation, where eight emotions are involved.



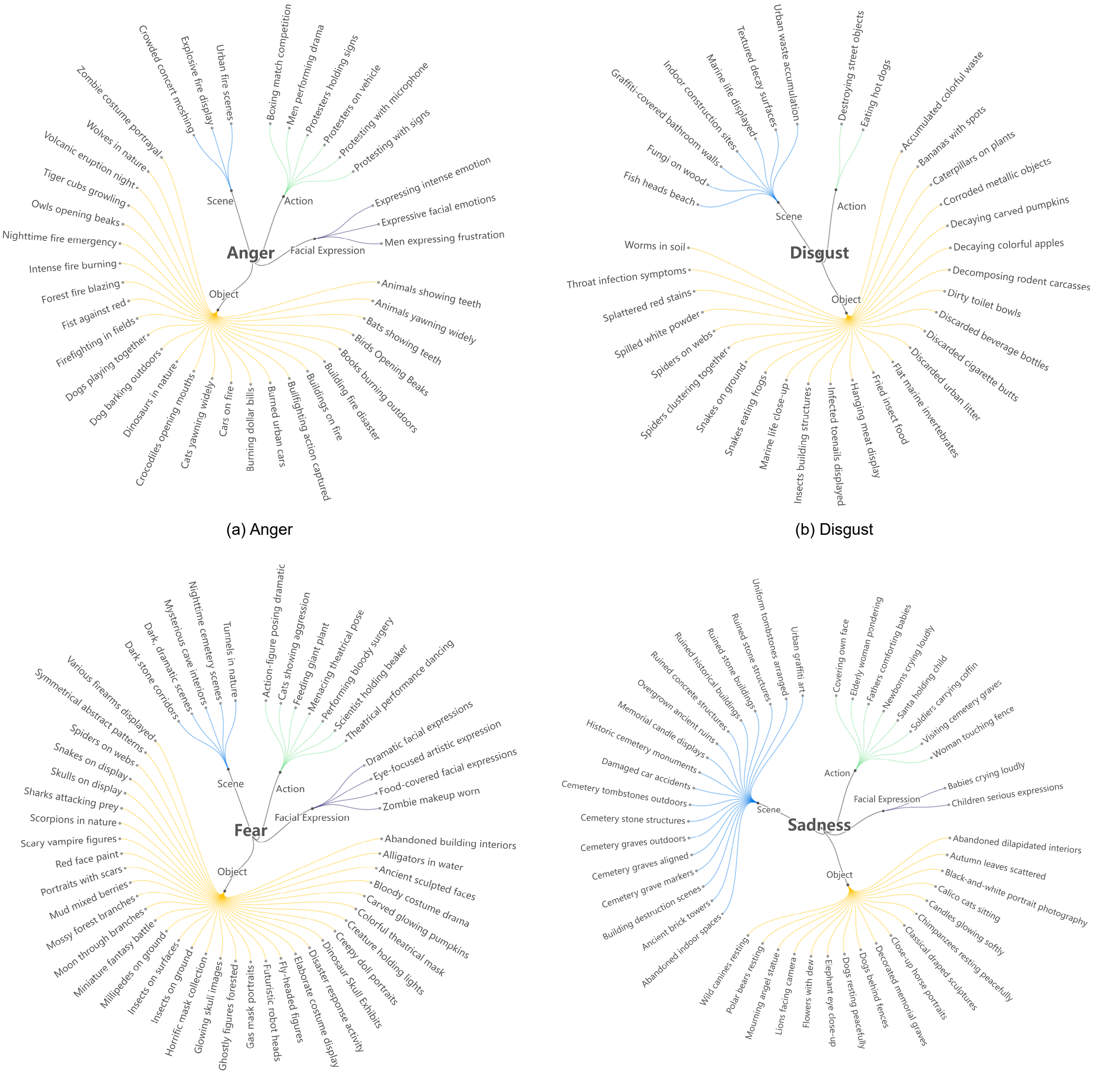
(a) Amusement

(b) Awe

(c) Contentment

(d) Excitement

Figure 11. Emotion factor tree on four *positive* emotions, comprising *amusement*, *awe*, *contentment* and *excitement*.



(a) Anger

(b) Disgust

(c) Fear

(d) Sadness

Figure 12. Emotion factor tree on four *negative* emotions, comprising *anger*, *disgust*, *fear* and *sadness*.