Algorithm 1 Map-Elites Algorithm

```
Initialize a map of solutions, each cell representing a unique feature combination
while not converged do
    Generate new solutions via mutation and crossover
    for each solution do
        Evaluate the solution for its performance and feature characteristics
        Identify the corresponding cell in the map based on features
        if solution is better than the current cell occupant then
            Replace the cell's solution with the new solution
        end if
    end for
end while
return the map of elite solutions
```

A Preliminaries on Quality Diversity

Quality Diversity (QD) (Mouret & Clune, 2015; Cully et al., 2015; Pugh et al., 2016; Lehman & Stanley, 2011b) is a concept in the field of optimization and artificial intelligence that emphasizes not just finding the best possible solution to a problem (quality), but also discovering a variety of good solutions that are diverse in their characteristics (diversity). This approach is particularly valuable in complex problem-solving scenarios where there might be multiple good solutions, each with its unique benefits.

A.1 Quality Diversity

The QD process aims to find the best representative samples, not only seeking the absolute best but also ensuring that the selections are varied and uniquely excellent in their own ways. Intuitively, imagine assembling a soccer team with QD: it meticulously recruits top-tier players across various positions to build a well-rounded team, rather than simply gathering the most renowned players regardless of their specialized roles. Key aspects of Quality Diversity include:

- Quality: This refers to how well a solution meets the desired criteria or objectives. In QD, the aim is to identify solutions that are highly effective or optimal with respect to the goals of the task.
- Diversity: Unlike traditional optimization that focuses on the single best solution, QD seeks a range of good solutions that are different from each other. This diversity can be in terms of features, approaches, or strategies the solutions employ.
- Exploration and Exploitation: QD balances exploration (searching for new, diverse solutions) and exploitation (refining known good solutions). This balance helps in thoroughly understanding the solution space and uncovering unique solutions that might be overlooked by conventional methods.

A.2 MAP-Elites

Map-Elites (Mouret & Clune, 2015) stands out in evolutionary computation for its unique approach to exploring solution spaces. Unlike traditional algorithms that target a single optimal solution, Map-Elites focuses on revealing a broad spectrum of high-performing solutions, categorized by distinct features. A high-level view of the Map-Elites algorithm, outlining its core steps, is shown in Algorithm 1.

The algorithm utilizes a grid or map where each cell corresponds to a unique combination of feature descriptors. New solutions are generated through mutation and crossover, and are evaluated for their performance and feature characteristics. The map is updated continually, with each cell holding the best-performing solution for its feature combination, ensuring a rich diversity of high-quality solutions.

Map-Elites is particularly advantageous in domains requiring adaptability and robustness, such as robotics, or in areas where creativity and a wide range of solutions are beneficial, like design and art.

Table 4: List of objective prompts for the LSI task.

Prompt 1	a photo of an astronaut riding a horse on mars
Prompt 2	an image of a bear in a national park
Prompt 3	an image of a cat on the sofa
Prompt 4	an image of a person playing guitar
Prompt 5	an image of a dog in the park
Prompt 6	an image of urban downtown

It provides insights into the solution space, highlighting the relationship between different solution features and their trade-offs.

B Additional Implementation Details

In this section, we detail our implementation and hyperparameters used in the experiments.

Robotic arm. The robotic arm repertoire task is configured to have 10 degrees of freedom, *i.e.*, the solution is a vector of 10 values, each specifying a joint angle. For QDHF and AURORA, we extract the features from the raw solution by running the accumulated sum on the solution vector and applying $\sin$ and $\cos$ transformations, resulting in a 20-dim feature vector. The latent projection in QDHF transforms the feature into a 2-dim embedding. For AURORA, the auto-encoder has an architecture of 64-32-2-32-64 neurons in each layer, where the mid-layer is the embedding used for QD. For QDHF, we use 1,000 judgments of simulated human feedback. The ground truth diversity is given by the end-point of the arm. For all experiments, we run Map-Elites for 1000 iterations, and for each iteration, we generate a batch of 100 solutions with Gaussian mutation (adding Gaussian noises sampled from $\mathcal{N}(0, 0.1^2)$), and evaluate them. The archive has a shape of $(50, 50)$, *i.e.*, each of the 2 dimensions is discretized into 50 equal-sized bins.

Maze navigation. For the maze navigation task, the solution is the network parameters of the default MLP policy network with a hidden-layer size of 8. The episode length of the environment is 250. We evaluate the policy and obtain the state descriptors. The objective is the accumulated reward at each state. For diversity measures, the ground truth diversity is the end-position of the agent, *i.e.*, the position at the last state. For QDHF and AURORA, we extract features from the state descriptor as the x and y positions of the agent at each state. The latent projection in QDHF transforms the feature into a 2-dim embedding. For AURORA, the auto-encoder has the same architecture of 64-32-2-32-64 nodes in each layer, where the mid-layer is the embedding used for QD. For QDHF, we use 200 judgments of simulated human feedback. For all experiments, we run Map-Elites for 1000 iterations, and for each iteration, we generate a batch of 200 solutions with Gaussian mutation (adding Gaussian noises sampled from $\mathcal{N}(0, 0.2^2)$), and evaluate them. The archive has a shape of $(50, 50)$.

Latent space illumination. In the LSI task, we run QDHF-online for 200 iterations with a batch size of 5 solutions per iteration. The solutions are generated with Gaussian mutation (adding Gaussian noises sampled from $\mathcal{N}(0, 0.1^2)$). The archive has a shape of $(20, 20)$. The solution is the latent vector used as the input to Stable Diffusion, which has a shape of $(4, 64, 64)$. We use Stable Diffusion v2.1-base, which generates images at a resolution of 512x512. The feature extractor is a CLIP model with ViT-B/16 backbone, which returns a 512-dim feature vector. QDHF learns a latent projection from 512-d to 2-d. To gather online human feedback, we use DreamSim with the DINO-ViT-B/16 backbone. The DreamSim model is trained on the NIGHTS dataset, which consists of 20k synthetic image triplets annotated with human judgments as labels. For QDHF, we use 10000 judgments of predicted human feedback. The objective is the CLIP score of the image and the text prompt. The text prompt is also input to the Stable Diffusion model to condition the generation towards more relevant content.

Table 5: Detailed quantitative results for the LSI task.

	Baseline			Baseline+			QDHF-Online		
#	CLIP	MeanPD	Std.PD	CLIP	MeanPD	Std.PD	CLIP	MeanPD	Std.PD
1	72.15	0.422	0.083	72.19	0.421	0.082	71.85	0.565	0.152
2	69.79	0.293	0.109	69.78	0.293	0.107	69.94	0.434	0.144
3	69.19	0.418	0.094	69.16	0.423	0.093	69.47	0.541	0.155
4	68.35	0.437	0.120	68.33	0.431	0.119	67.95	0.571	0.147
5	68.90	0.564	0.117	68.97	0.569	0.119	69.43	0.603	0.167
6	64.74	0.388	0.113	65.01	0.382	0.113	65.83	0.448	0.140

Table 6: Detailed user study results for the LSI task. The survey data is collected on 43 participants through an online form.

	User Preference			Perceived Diversity		
	Baseline	QDHF	Hard to Decide	Baseline	QDHF	Hard to Decide
1	22	17	4	2	40	1
2	12	21	10	2	33	8
3	6	30	7	3	36	4
4	13	28	2	7	34	2
5	6	28	9	5	23	15
6	10	17	16	4	28	11

C ADDITIONAL RESULTS FOR LATENT SPACE ILLUMINATION

We include more detailed results on the LSI task with different prompts. Table 4 lists the prompts that we used as objectives for the LSI task. Table 5 includes detailed quantitative results for each prompt. Table 6 includes detailed user study results for each prompt, which summarizes data from a survey of 43 participants with varying experience with text-to-image software. A sample question used in the user feedback survey is shown in Fig. 4. We also show examples of generated images in Fig. 5 to Fig 10. For all the figures, the top 4x4 grid displays images with the highest CLIP scores from randomly generated images. The bottom grid displays a uniformly-sampled subset of QDHF-online solutions. Qualitatively, images generated by QDHF have more variations, and show visible trends of diversity.

Notably, while QDHF significantly outperforms the baseline on most cases, we find that prompt 5 and 6 are two sub-performing cases for QDHF. For both prompts, the diversity of QDHF results is not apparent. The most likely reason is that the preference model (DreamSim) does not generalize well to cases such as different breeds of dogs, scenes of a park, and appearances of cityscapes. We aim to solve the above issues in future work where human feedback needs to be collected in a more diverse and strategic way to facilitate better generalization of the preference model, and thus improves the performance of QDHF. Another interesting finding is that for prompt 1, while most users find QDHF results are more diverse, more than half of the users actually prefers the less diverse baseline results. According to the feedback from users, people may prefer less diverse but more content-focused results in some specific cases. The relationship between diversity and user preference under different use cases in generative AI application remains an open question, and we look forward to exploring this topic in future work.

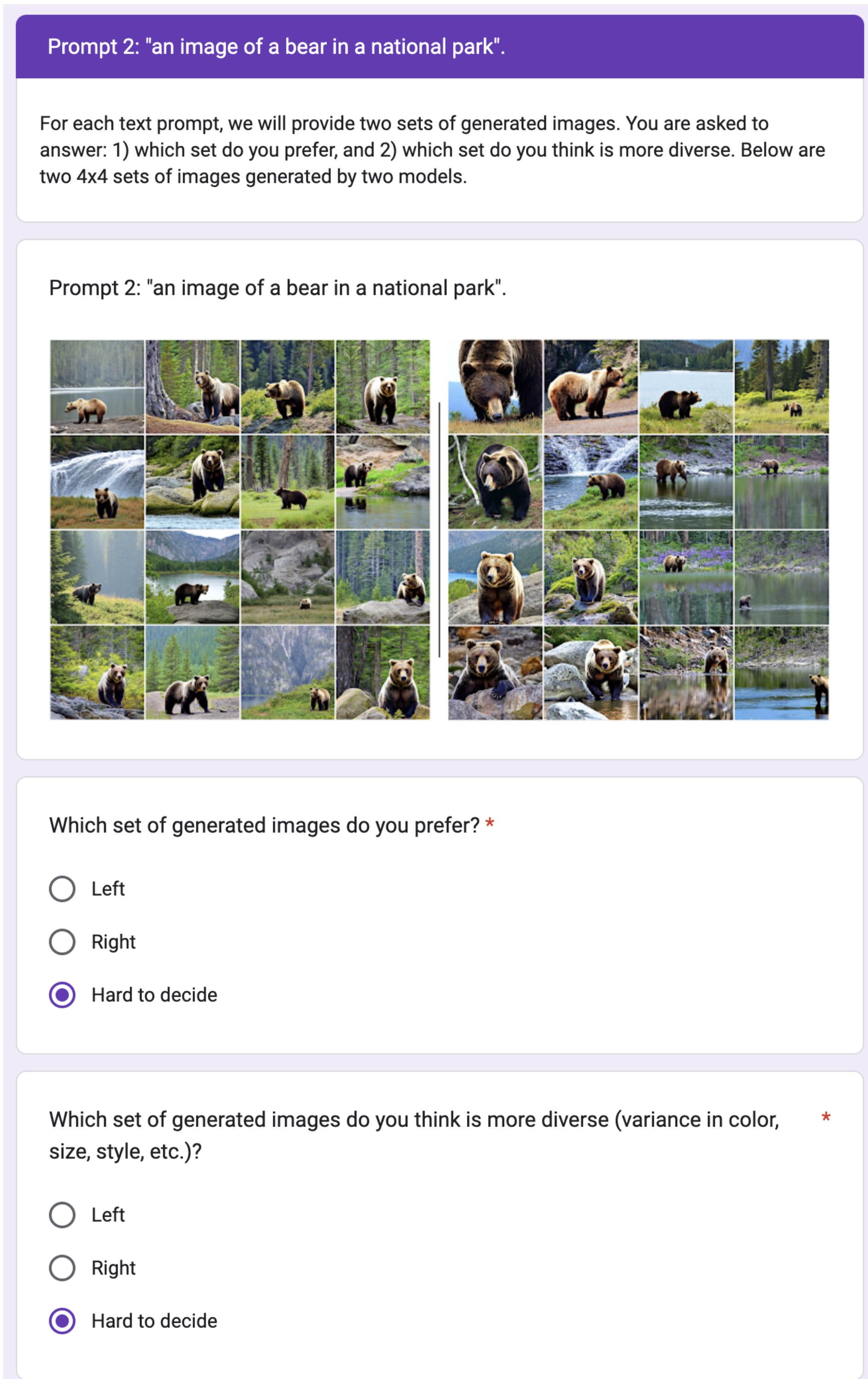

Figure 4: Sample question used in the user feedback survey.

Figure 5: Qualitative results for prompt 1: “a photo of an astronaut riding a horse on mars”. The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.

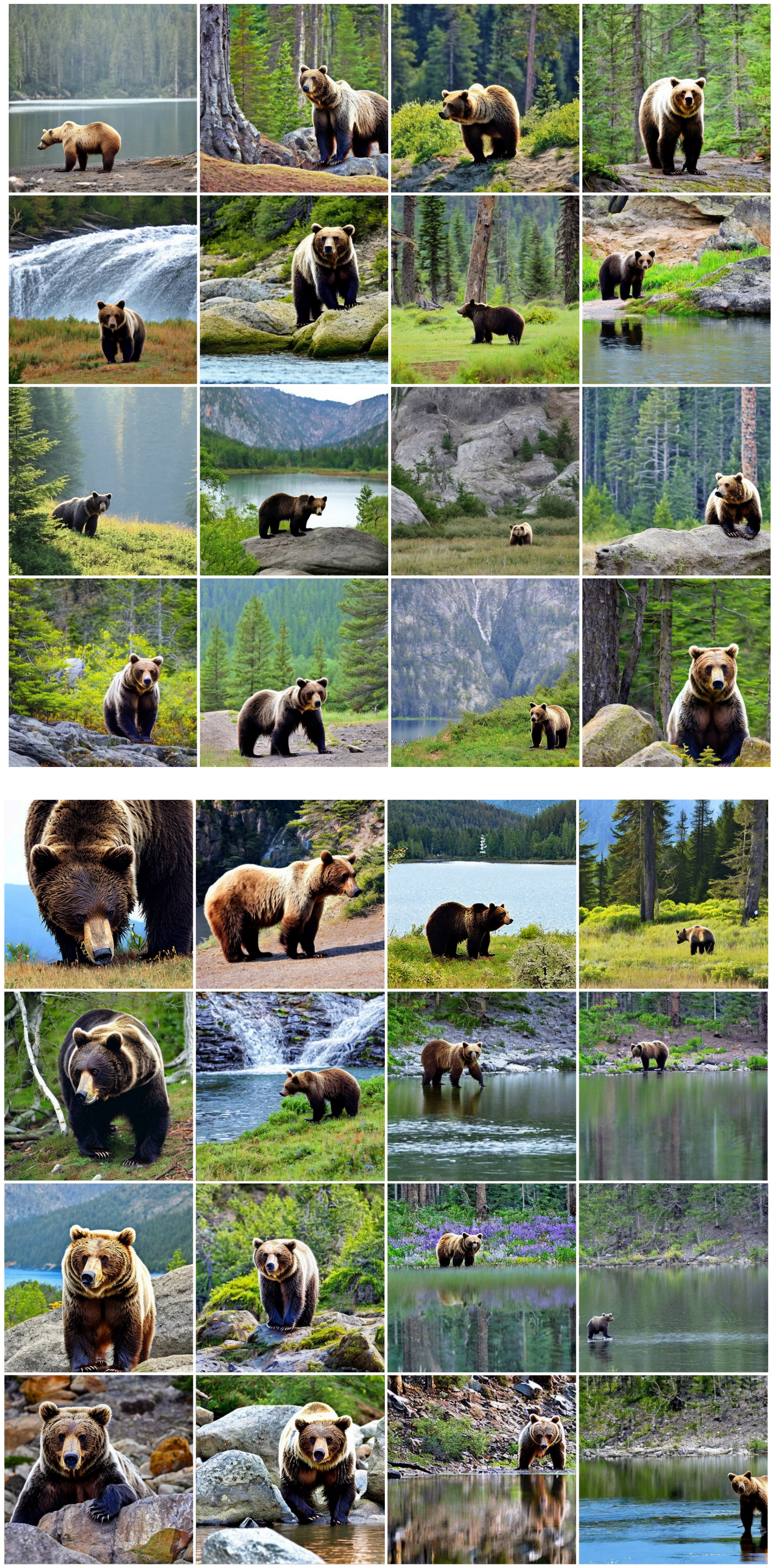

Figure 6: Qualitative results for prompt 2: “an image of a bear in a national park”. The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.

Figure 7: Qualitative results for prompt 3: “an image of a cat on the sofa”. The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.

Figure 8: Qualitative results for prompt 4: “an image of a person playing guitar”. The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.

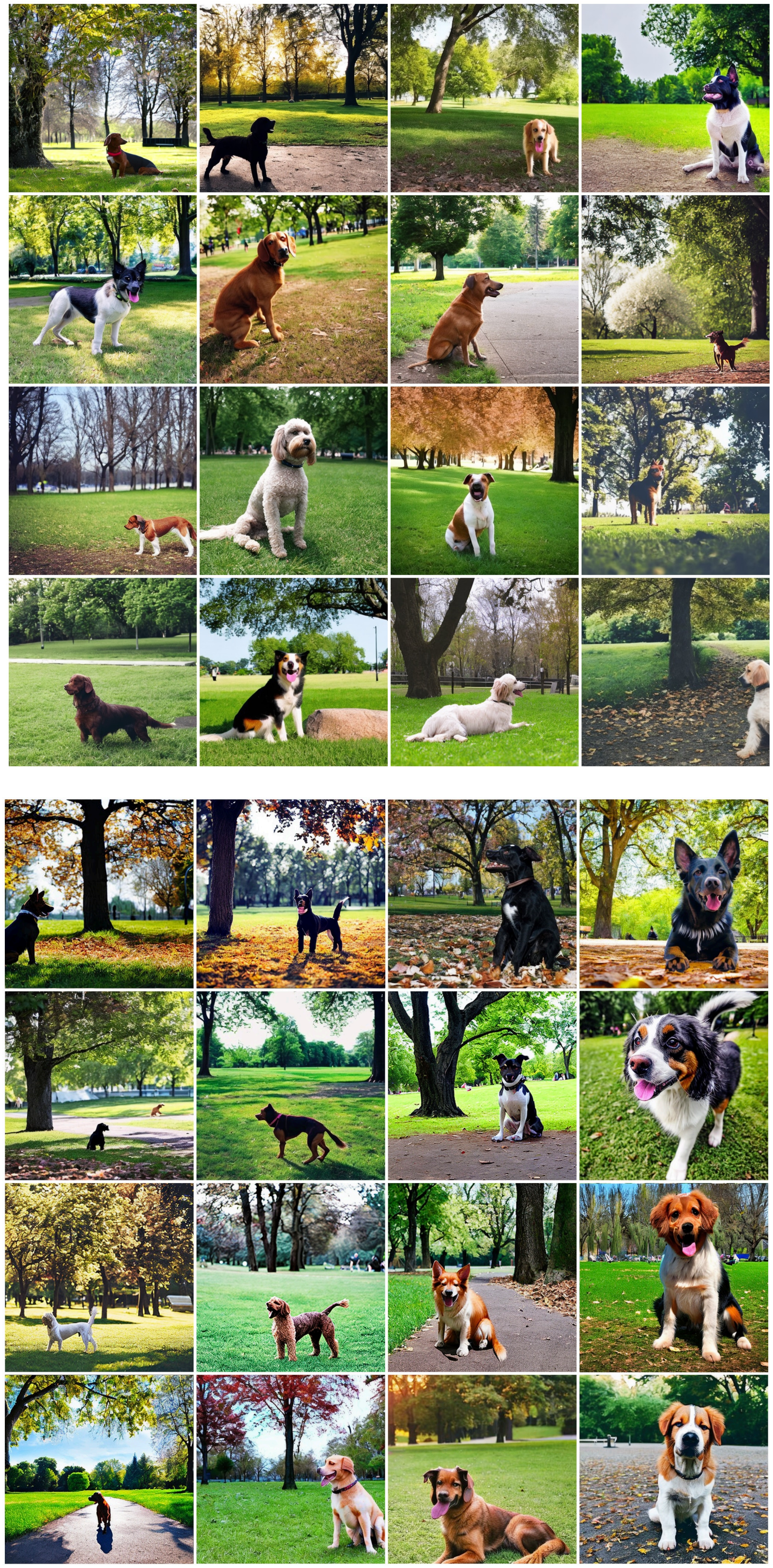

Figure 9: Qualitative results for prompt 5: "an image of a dog in the park". The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.

Figure 10: Qualitative results for prompt 6: “an image of urban downtown”. The top 4x4 grid shows the baseline results and the bottom grid shows QDHF results.