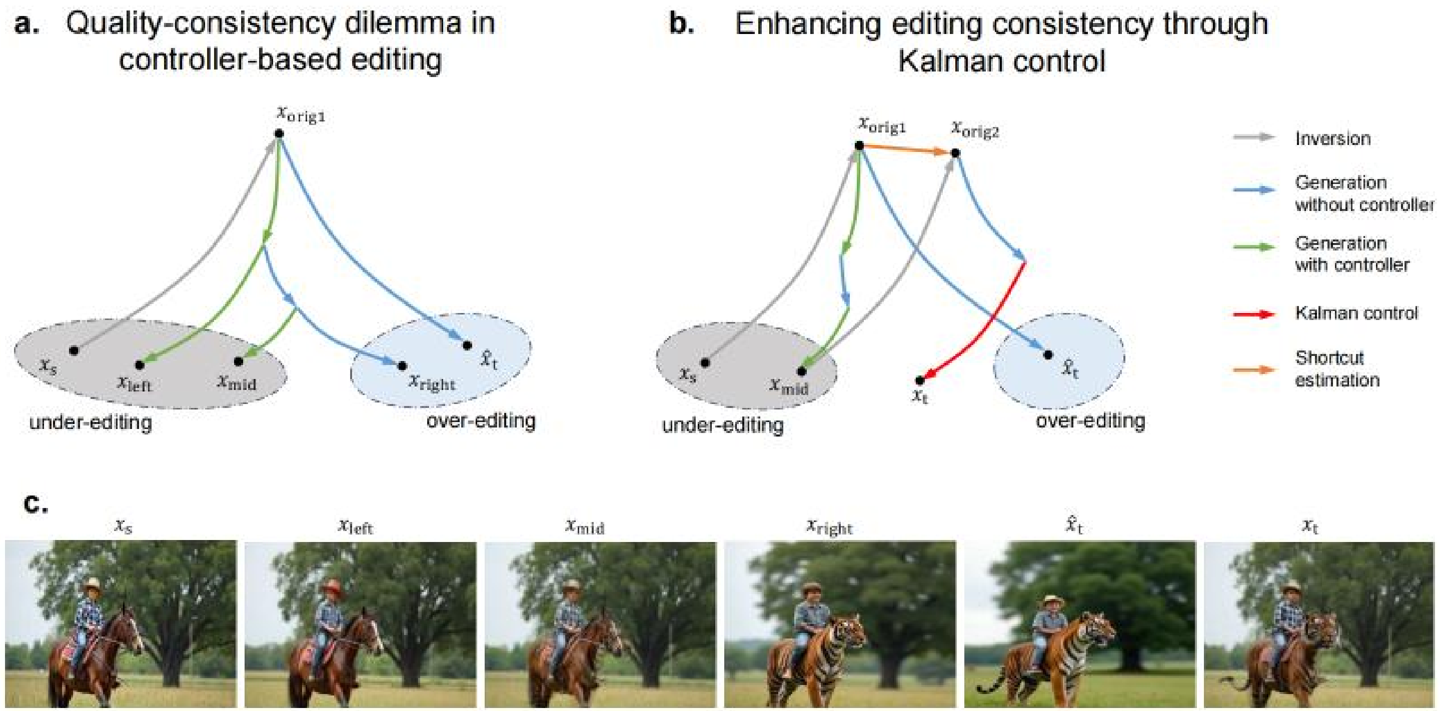
Enhancing Consistency of Flow-Based Image Editing through Kalman Control

Haozhe Chi 1, Zhicheng Sun 1, Yang Jin 1,
Yi Ma 2, Jing Wang 2, Yadong Mu 1

1 Peking University, 2 Central Media Technology Institute, Huawei

Introduction

- Motivation:
- * Many diffusion-based editing methods still face challenges related to imprecise editing, primarily due to the non-linear nature of the generation trajectory.
- * Image editing methods in flow-based models, such as controller-based method, exhibit limitations in editing quality, highlighting the need for further refinement in this area.



Introduction

- Main contributions:
  - We introduce the Kalman filter approach to controller-based image editing.
  - Based on observations of direct LQR control, we propose a two-stage method that better unleashes the potential of Kalman control and achieves more flexible control.
  - Through experiments, our method shows high structural consistency and good editing flexibility on various editing tasks.

Preliminaries

Optimal control. To improve editing controllability, an effective approach is to formulate it as optimal control [43]. Both inversion and generation processes can be viewed as a continuous-time linear system defined over t in $[0, 1]$.

$$\frac{dx_t}{dt} = Ax_t + Bu_t, \tag{4}$$

where x_t represents the state of the system, u serves as the controller of the system, and A, B are coefficient matrices. In optimal control theory, the objective is to determine an optimal controller u_t that guides the drift path to minimize the energy cost. A choice of the energy cost is a quadratic function, which corresponds to the Linear Quadratic Regulator (LQR) problem [21] as follows:

$$J_1 = x_1^\top F x_1 + \int_0^1 \left(x_t^\top Q x_t + u_t^\top R u_t \right) dt, \tag{5}$$

where F, Q and R are coefficient matrices of the system. As shown in previous work [43], solving the problem for rectified flow in Eq. (2) produces the optimal controller:

$$u_t = \frac{x_s - x_t}{1 - t}, \tag{6}$$

Method: Kalman control for LQG problem

$$\frac{dx_t}{dt} = Ax_t + Bu_t + w_t, \quad y_t = Hx_t + \sigma_t, \tag{7}$$

where y_t represents the measurement sequence, A, B and H are the coefficient matrices of the system, while w_t and σ_t denote the noise terms of system state estimations. With the application of the Kalman filter, the total cost function to be minimized is given by:

$$J_2 = \mathbb{E}\left[x_1^\top F x_1 + \int_0^1 \left(x_t^\top Q x_t + u_t^\top R u_t\right) dt\right], \tag{8}$$

where $\mathbb{E}$ refers to the expectation of the following terms. Importantly, expectation is necessary in this context, as our goal is to mitigate the impact of noise terms when minimizing the cost function J_2. To accurately estimate the expectation in J_2, we utilize the following Kalman filter equations:

$$\begin{aligned} K_k &= P_{k-1}H^T(HP_{k-1}H^T + T)^{-1}, \\ x_k &= Ax_{k-1} + Bu_k + K_k(y_k - Hx_{k-1}), \\ P_k &= (I - K_kH)P_{k-1}. \end{aligned} \tag{9}$$

Method: Practical algorithm

Algorithm 1 **Kalman-Edit and Kalman-Edit***

Input: original image x_{s}, total step N, source prompt c_{s}, target prompt c_{t}, Inversion function $\mathrm{Inv}(\cdot,\cdot)$

 # Stage 1: editing with optimal controller

$x_{\mathrm{orig1}} \leftarrow \mathrm{Inv}(x_{\mathrm{s}}, N)$

Generate x_{mid} from x_{orig1} with controller in Eq. (6)

 # Stage 2: editing with Kalman filter

if Kalman-Edit **then**

$x_{\mathrm{orig2}} \leftarrow \mathrm{Inv}(x_{\mathrm{mid}}, N)$

else if Kalman-Edit* **then**

$x_{\mathrm{orig2}} \leftarrow x_{\mathrm{orig1}} + x_{\mathrm{mid}} - x_{\mathrm{s}}$ ▷ shortcut in Section 3.4

Compute measurement $\{y_k\}_{k=1}^{l}$ according to Eq. (11)

Generate x_{t} from x_{orig2} with Kalman filter in Eq. (12)

Output: x_{t}

Quantitative results

Table 1: Quantitative comparison on SFHQ datasets among flow-based editing models. See the main text for the definitions of the performance metrics. The highest value in each column is highlighted in bold.

	Face Rec. ↓	CLIP-I ↑	LPIPS ↓	CLIP-T ↑	DreamSim ↓
RF-Edit	0.4051	0.8984	0.1562	0.2910	0.1591
RF-Inversion	0.4325	0.8927	0.1720	**0.3012**	0.1889
FlowEdit	0.4856	0.8579	0.1687	0.2905	0.2375
FlowChef	0.4013	0.8769	0.1401	0.2832	0.1487
Kalman-Edit	**0.3958**	**0.9167**	**0.1332**	0.2921	**0.1408**
Kalman-Edit*	0.4696	0.8871	0.1892	0.2936	0.2227

Table 2: Quantitative comparison on HQ datasets among flow-based editing models.

	CLIP-T ↑	CLIP-I ↑	LPIPS ↓	DINO ↑	DreamSim ↓
RF-Edit	0.1842	**0.9141**	0.2383	**0.8197**	0.1492
RF-Inversion	0.1825	0.9033	0.3074	0.7963	0.1662
FlowEdit	0.1877	0.8813	0.2846	0.7467	0.2238
FlowChef	0.1928	0.9023	0.2925	0.8053	0.1537
Kalman-Edit	**0.1943**	0.9062	**0.2345**	0.7929	**0.1353**
Kalman-Edit*	0.1870	0.8696	0.3615	0.7123	0.2276

Quantitative results

Table 3: Quantitative evaluations on ZONE and DIV2K datasets. See main text for more details about the performance metrics.

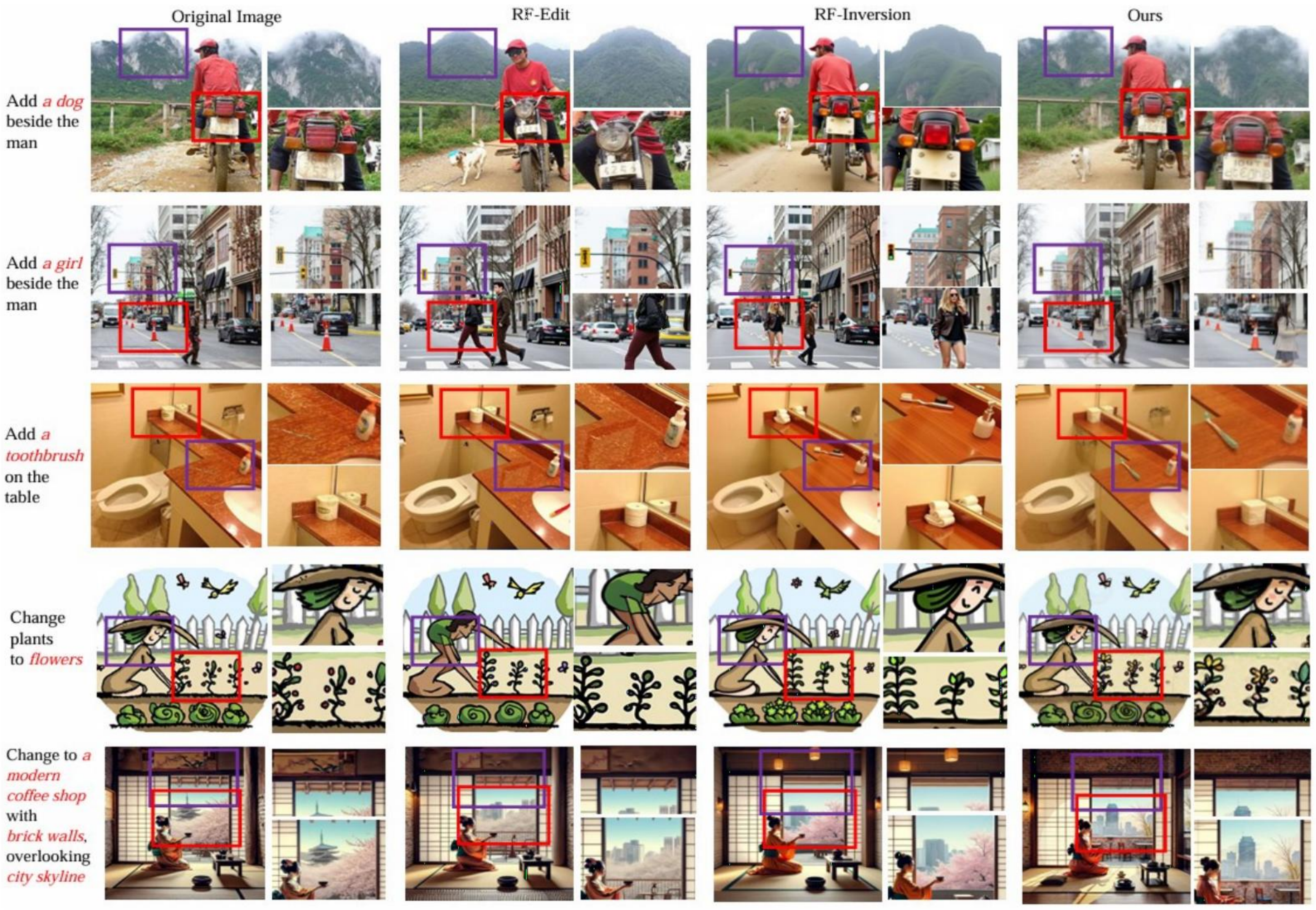
	CLIP-T ↑	CLIP-I ↑	LPIPS ↓	DINO ↑	DreamSim ↓
SDEdit	0.2754	0.9264	0.1908	0.8547	0.1148
P2P	0.2773	0.9209	0.1568	0.8186	0.1519
MasaCtrl	0.3103	0.9179	0.1580	0.8397	0.1635
DDPM-Inv	0.2847	0.9063	0.1734	0.8215	0.1742
RF-Edit	0.2964	0.8926	0.2039	0.7986	0.1776
RF-Inversion	0.2844	0.8919	0.2491	0.7974	0.1536
FlowEdit	0.3096	0.8687	0.2269	0.7671	0.2319
FlowChef	0.3025	0.8831	0.2563	0.7456	0.2471
Kalman-Edit	0.2957	**0.9492**	**0.1407**	**0.9141**	**0.0793**
Kalman-Edit*	**0.3220**	0.8986	0.2488	0.8237	0.1454

Table 4: Comparison of CLIP-I (left) and LPIPS scores(right) for different Kalman filter strengths and steps evaluated on the ZONE dataset.

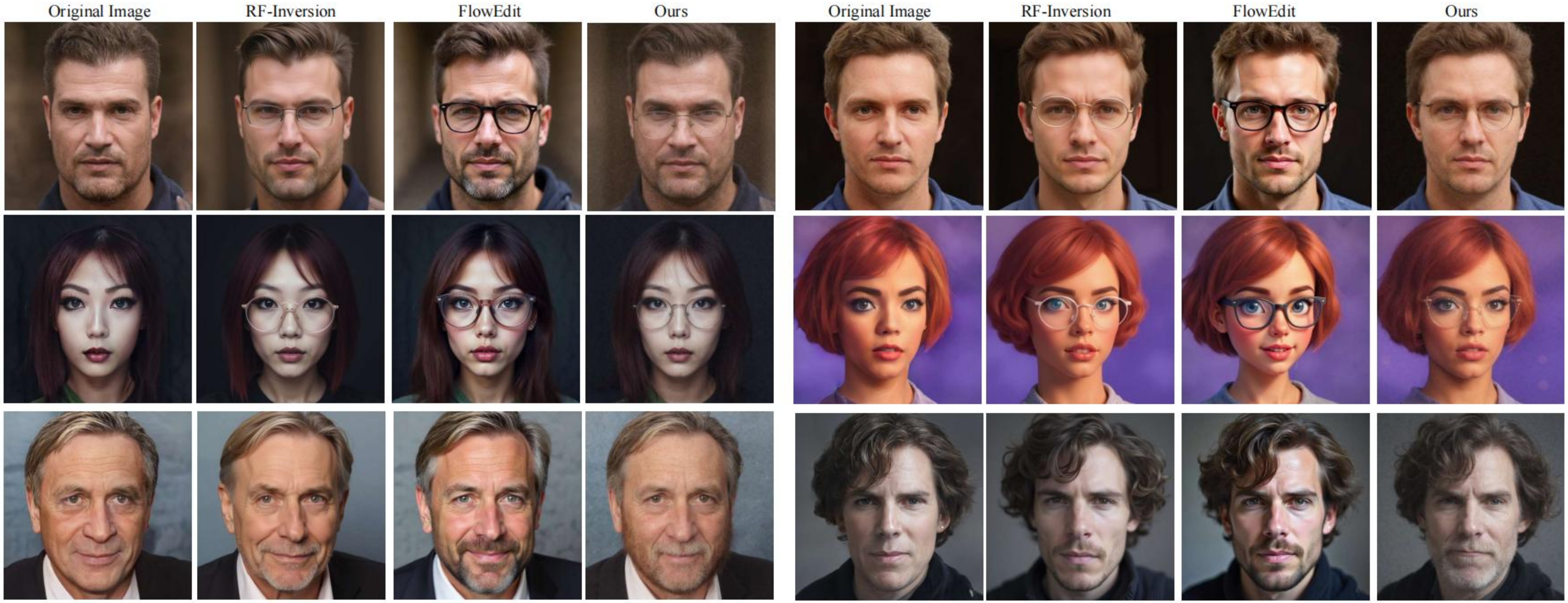
Filter strength / Added steps	15-18	15-22	15-27
0.1	0.8770	0.9219	**0.9346**
0.2	0.9043	0.9282	0.9226
0.3	0.9014	0.8921	0.9079

Filter strength / Added steps	15-18	15-22	15-27
0.1	0.2433	0.2035	**0.1487**
0.2	0.2325	0.1944	0.1521
0.3	0.2284	0.2008	0.1886

Qualitative results



Qualitative results



More editing cases

[Simple and elegant style] an empty room with wooden floor and a view of the *[impressionism beach]*

Change the *beach* in the background to a *snow mountain*

Change the castle to *Disney cartoon style* and add *a bridge*

Add *fireworks*

Lizard ===> *Dragon*

Horse ===> *Camel*

Add *a boat*

Add *steak*

Panda ===> *Racoon*

Lion ===> *Tiger*

Duck ===> *Swan*

Add *aurora*

Add *a balloon*

Lion ===> *Wolf*

Rock ===> *Branch*

Thanks for watching