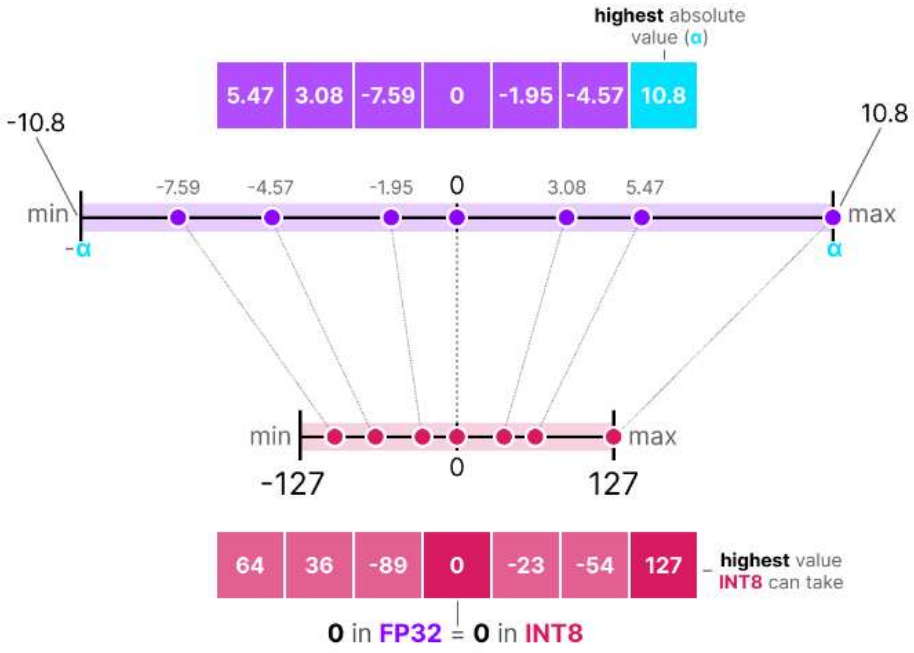
SageAttention

(https://arxiv.org/pdf/2410.02367)

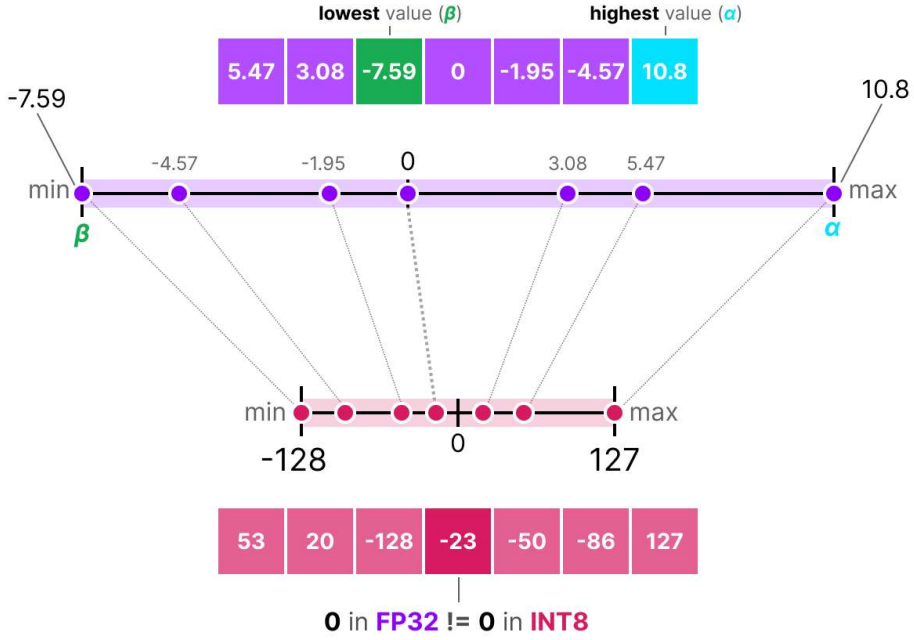
Виды квантизации

Симметричная квантизация:



$$X_{INT8} = round\left(\frac{X_{FP16}}{scale^X}\right)$$

Асимметричная квантизация:



$$X_{INT8} = round\left(\frac{X_{FP16}}{scale^X}\right) + offset^X$$

Виды квантизации

Квантизация по **токенам** (per-token quantization)

$scale^X$
0.01
0.05
0.06

X_{FP16}		
1.47	-0.38	**-1.63**
1.54	2.14	**6.63**
8.82	-8.00	5.84

X_{INT8}		
114	-30	**-127**
30	41	**127**
127	-115	84

Формула **симметричной** квантизации (symmetric quantization)

$$X_{INT8} = round\left(\frac{X_{FP16}}{scale^X}\right) \qquad scale^X = \frac{\max(|X_{FP16}|)}{127}$$

$$\hat{X}_{FP16} = X_{INT8} \cdot scale^X$$

Квантизация по **каналам** (per-channel quantization)

$scale^X$		
0.04	0.24	0.06

X_{FP16}		
3.02	0.05	2.14
6.17	1.2	**-8.00**
-3.82	**31.02**	1.67

X_{INT8}		
62	0	34
127	5	**-127**
-79	**127**	27

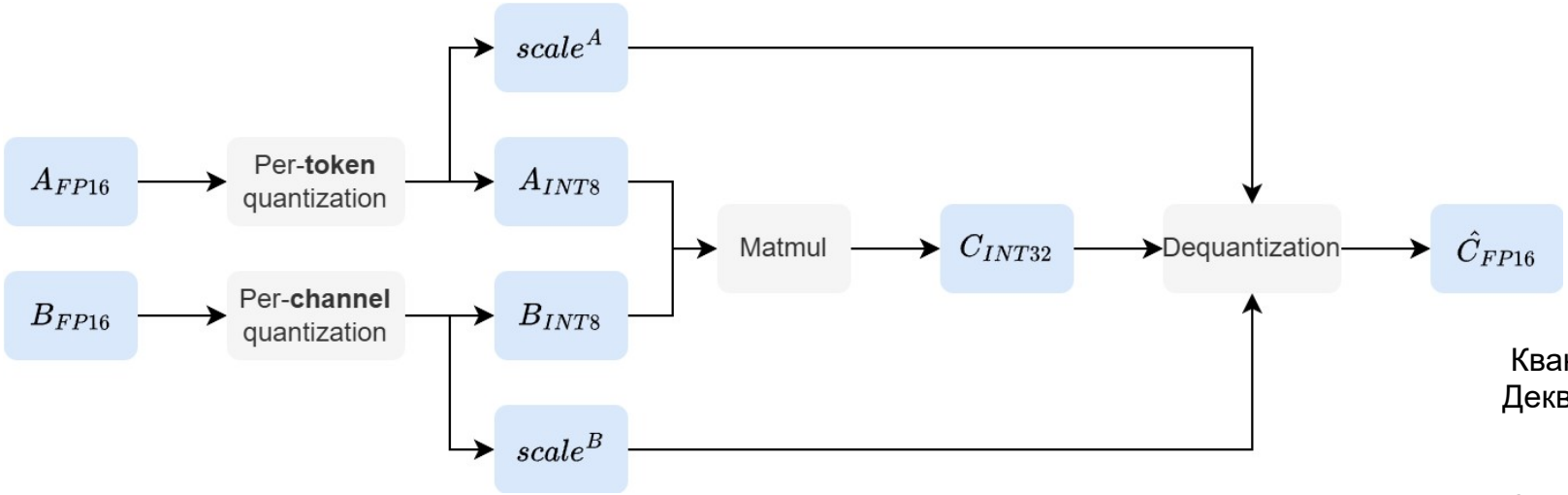
Квантизация по **блокам** (per-block quantization)

$scale^X$
0.05
0.07

X_{FP16}			
-0.28	1.47	-0.38	-1.63
3.02	1.54	2.14	**6.63**
6.17	8.82	-8.00	5.84
-3.82	**-9.65**	1.67	0.04

X_{INT8}			
-6	28	-7	-31
58	30	41	**127**
81	116	-105	77
-50	**-127**	22	1

Матричное умножение квантизованных матриц



Квантизация / Деквантизация

$O(C_1 n^3 + C_2 n^2)$

Матричное умножение

$$A_{INT8} = round\left(\frac{A_{FP16}}{scale^A}\right) = (a'_{ij}) \qquad C_{INT32} = (c'_{ij}), \quad c'_{ij} = \sum_k a'_{ik} \cdot b'_{kj}$$

$$B_{INT8} = round\left(\frac{B_{FP16}}{scale^B}\right) = (b'_{ij})$$

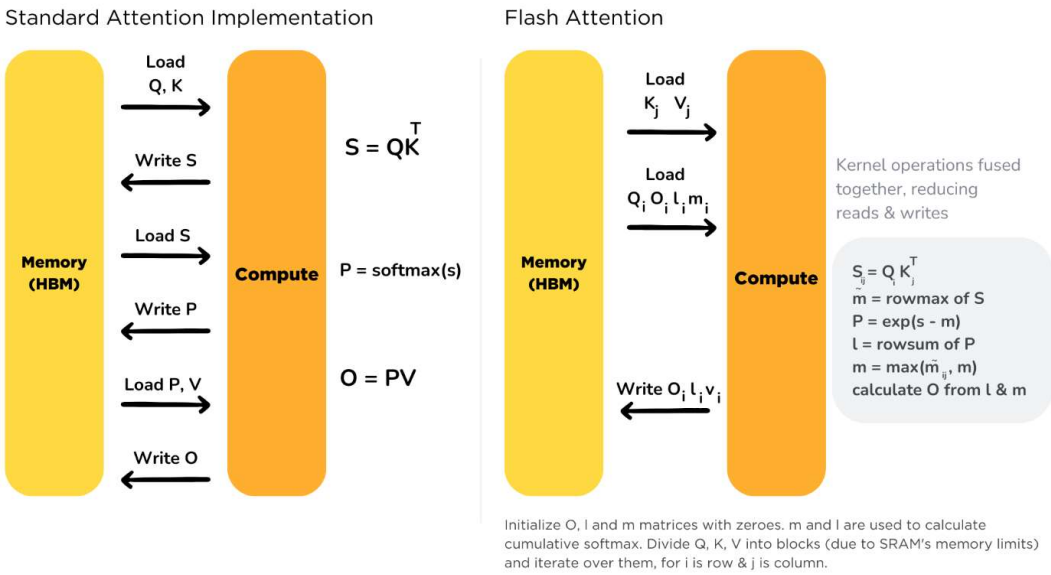
Если A_{FP16} квантизуется по **токенам**, а B_{FP16} – по **каналам**, то $\boldsymbol{ind_1} = i$ и $\boldsymbol{ind_2} = j$:

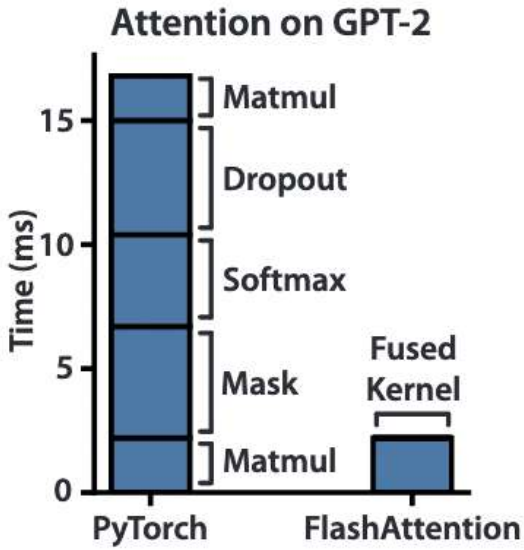
$$\hat{C}_{FP16} = (c_{ij}), \quad c_{ij} = \sum_k a'_{ik} \cdot scale^A_{\boldsymbol{ind_1}} \cdot b'_{kj} \cdot scale^B_{\boldsymbol{ind_2}} = \sum_k scale^A_{\boldsymbol{i}} \cdot scale^B_{\boldsymbol{j}} \cdot a'_{ik} \cdot b'_{kj} = scale^A_{\boldsymbol{i}} \cdot scale^B_{\boldsymbol{j}} \cdot c'_{ij}$$

Attention, FlashAttention

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V$$

- FlashAttention достигает ускорения за счет оптимального управления памятью и использования высокоэффективного CUDA-ядра (основанного на CUTLASS). В нем несколько стадий вычислений объединены в единую операцию, что минимизирует обращения к HBM и снижает накладные расходы.
- **Идея SageAttention**: дополнительно ускорить FlashAttention посредством целочисленного низкобитного матричного умножения вместо вещественного.



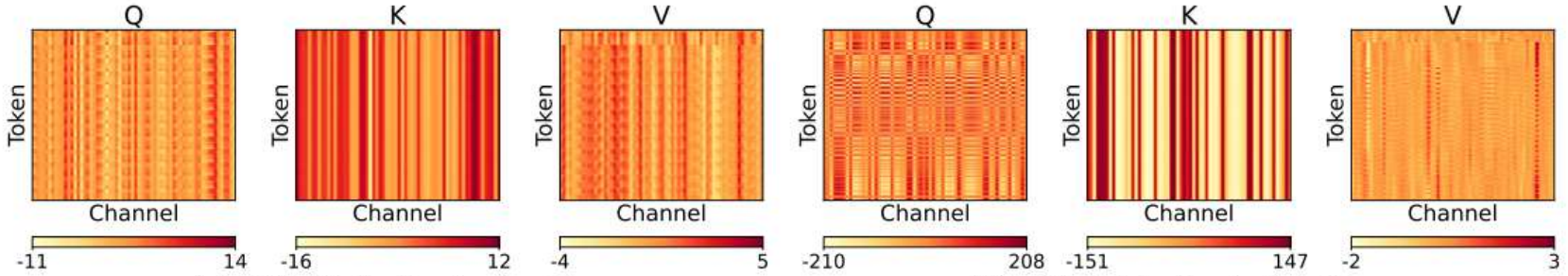


SageAttention: выбросы в K

- **Проблема**: матрица K содержит большие значения (выбросы), сконцентрированные по каналам матрицы, обычная квантизация приводит почти к полному обнулению остальных элементов.
- **Решение**: вычесть из матрицы K **среднее** по каждому столбцу.

$$QK^T = (p_{ij}), \quad p_{ij} = \sum_l q_{il} \cdot (k_{il} - \bar{k}_i) = \sum_l q_{il} k_{il} - \sum_l q_{il} \bar{k}_i = \sum_l q_{il} k_{il} - C_i$$

$$softmax([p_{i1}, p_{i2}, \dots, p_{in}]) = \frac{e^{-C_i} \cdot [e^{p_{i1}}, e^{p_{i2}}, \dots, e^{p_{in}}]}{e^{-C_i} \cdot \sum_k e^{p_{ik}}} = softmax([p_{i1}, p_{i2}, \dots, p_{in}] - C_i)$$



(a) QKV distribution in Cogvideo (b) QKV distribution in Unidiffuser

Строка матрицы K

0.05	31.02	1.2	…	0.7

Без вычитания **среднего**:

0	127	5	…	3

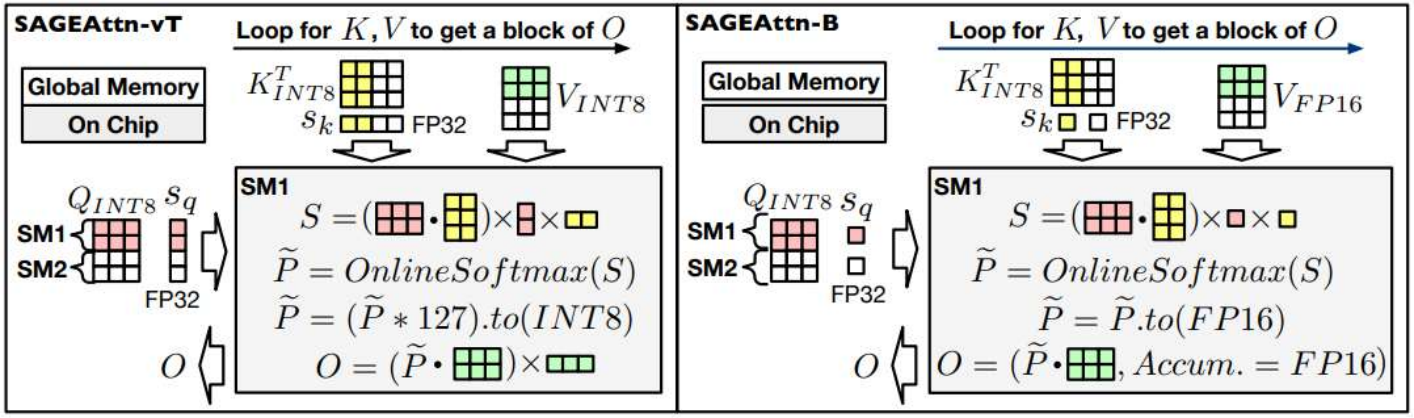
С вычитанием **среднего**:

3	127	75	…	44

SageAttention

1. Вычесть из матрицы K среднее: $K = K - mean(K)$, матрицу Q поделить на число $\sqrt{d_k}$;
2. Квантизовать матрицы Q и K по блоку токенов $Q_{INT8},\ S_q = per\_block(Q),\ K_{INT8},\ S_k = per\_block(K)$
3. Выполнить первое (целочисленное) матричное умножение: $\hat{P}_{FP16} = (Q_{INT8}K_{INT8}^T)_{INT32} \cdot S_q \cdot S_k$;
4. Вычислить softmax: $P = softmax(\hat{P}_{FP16})$;
5. Выполнить второе (вещественное) матричное умножение: $P \cdot V$

* *Второе матричное умножение также можно сделать целочисленным, однако это не приносит значительного ускорения, а качество работы значительно ухудшается.*



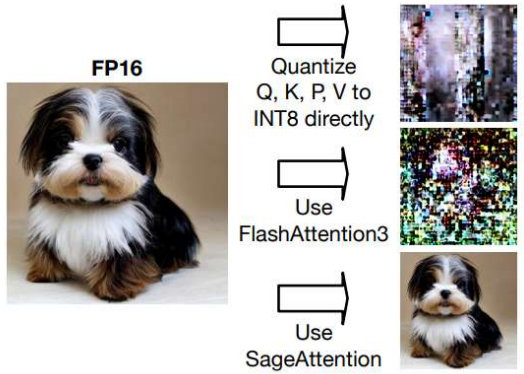
(a) SageAttention (per-token quantize Q,K; INT8 V) (b) SageAttention (per-block quantize Q,K; FP16 V)

SageAttention: результаты

- SageAttention обеспечивает более чем **двукратное ускорение** по сравнению с FlashAttention;
- Ускорение сильно зависит от длины последовательности: чем длиннее контекст, тем большую часть времени занимает Attention, и тем больше выигрыш от низкобитного целочисленного матричного умножения;
- Максимальный прирост достигается в диффузионных моделях, где Attention операция занимает примерно 85-90% всего инференса. Так, модель CogVideoX работает в 1.35 раз быстрее за счет SageAttention;
- Качество моделей остается почти неизменным, благодаря коррекции выбросов;

Table 7: Real speedup of `SageAttention` on RTX4090.

Model	Shape of Q, K, V	Original attention	SageAttention	Speedup
`CogvideoX`	(2, 30, 17776, 64)	163.37 (FlashAttn2)	**327.57**	**2.01x**
`Llama2`	(4, 32, 1536, 128)	130.99 (FlashAttn2)	**231.74**	**1.77x**
`UltraPixel`	(2, 32, 7285, 64)	152.03 (FlashAttn2)	**325.18**	**2.14x**
`Unidiffuser`	(4, 24, 1105, 64)	105.68 (xformers)	**246.93**	**2.34x**
`TIMM`	(12, 64, 197, 64)	18.910 (Torch)	**111.41**	**5.89x**



SageAttention: результаты

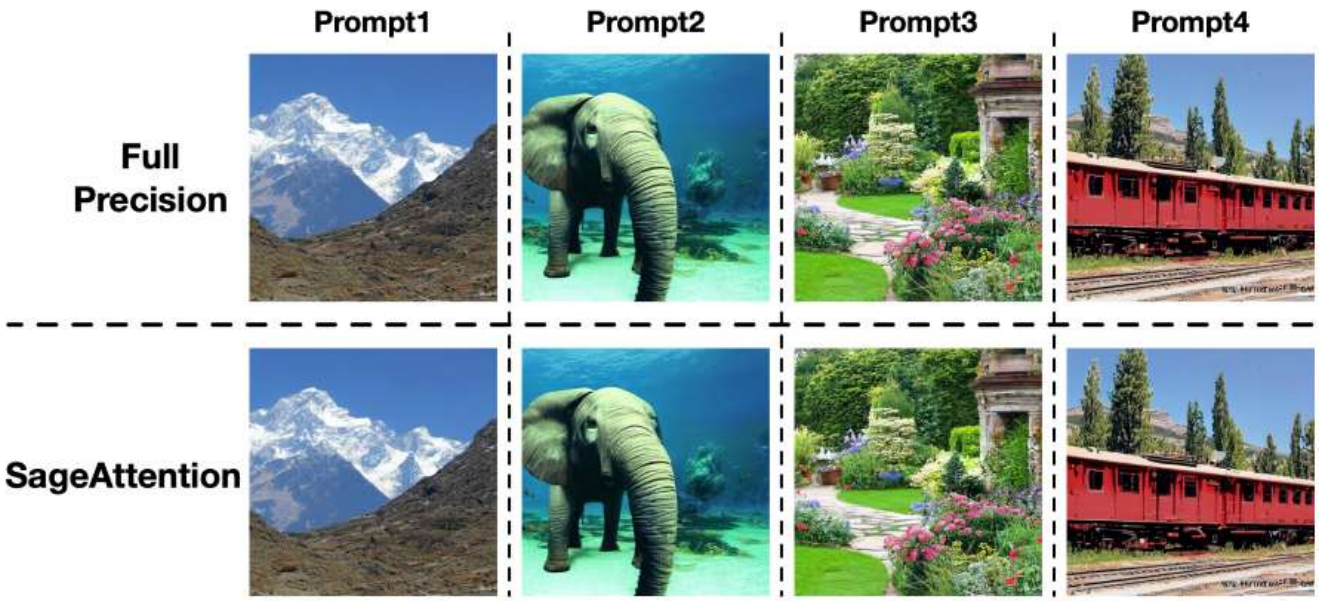


Figure 13: More image generation examples of Unidiffuser, where prompt1=”Beautiful view of the Himalayas”, prompt2=”An elephant under the sea”, prompt3=”English Country Garden Design”, and prompt4=”An old red electric rail train in Durango, Colorado”.

SageAttention: результаты

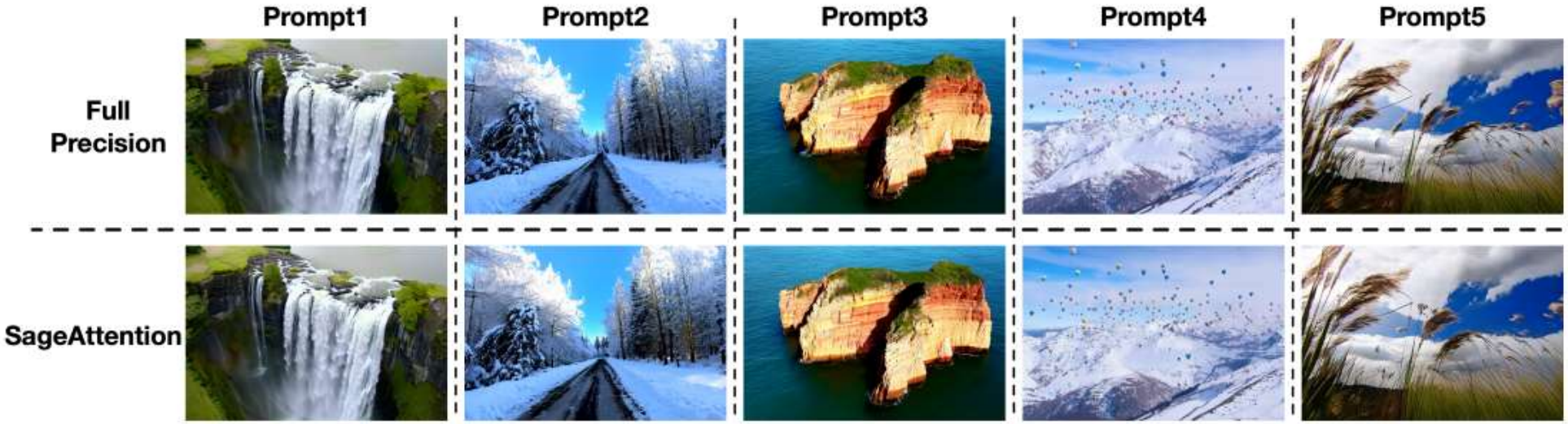


Figure 14: More image generation examples of CogvideoX. For more details about the prompts and the full videos, refer to `https://anonymous.4open.science/r/image_video_examples-3E44/README.md`.

Спасибо за внимание!