

IT/IST/IPLeiria Report on JPEG PCC Core Experiment 4.2: Sparse Convolutions

Doc. WG1m100113

André Guarda, Nuno Rodrigues, Fernando Pereira

*Instituto de Telecomunicações, Instituto Superior Técnico, and Instituto Politécnico de Leiria
PORTUGAL*

July 2023

Outline

- 1. Context and Objective**
- 2. Sparse Convolutions in the VM**
- 3. Performance Assessment: Sparse vs Dense Convolutions**
 - Evaluating Super-Resolution Models
 - Evaluating Overall Codec on JPEG CTTC
 - Evaluating Overall Codec on Sparse MPEG PCs
- 4. Sparse Convolutions in the VM: Pruning**
- 5. Conclusions**



instituto de
telecomunicações



Context and Objective

Context and Objective

- ★ **The current VM uses a dense 3D representation with dense convolutions**
- ★ **Problem:**
 - Heavy in computation complexity, cannot encode a full point cloud at once
 - Underperforms for sparse point cloud
- ★ **Objective:**
 - Implement the DL models in the VM with a sparse tensor representation, and verify its performance



Sparse Convolutions in the VM

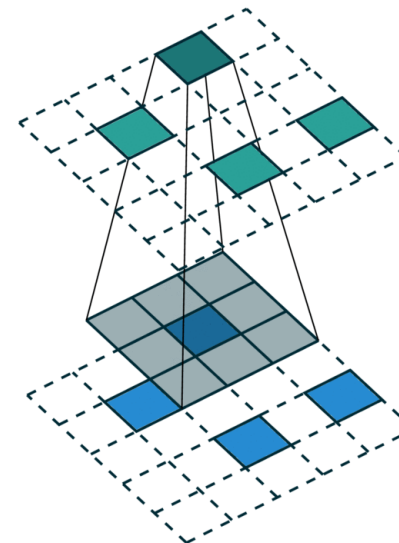
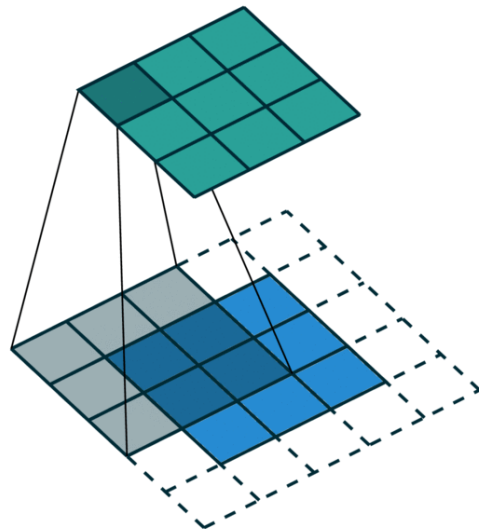
Dense vs Sparse Convolutions

★ Dense:

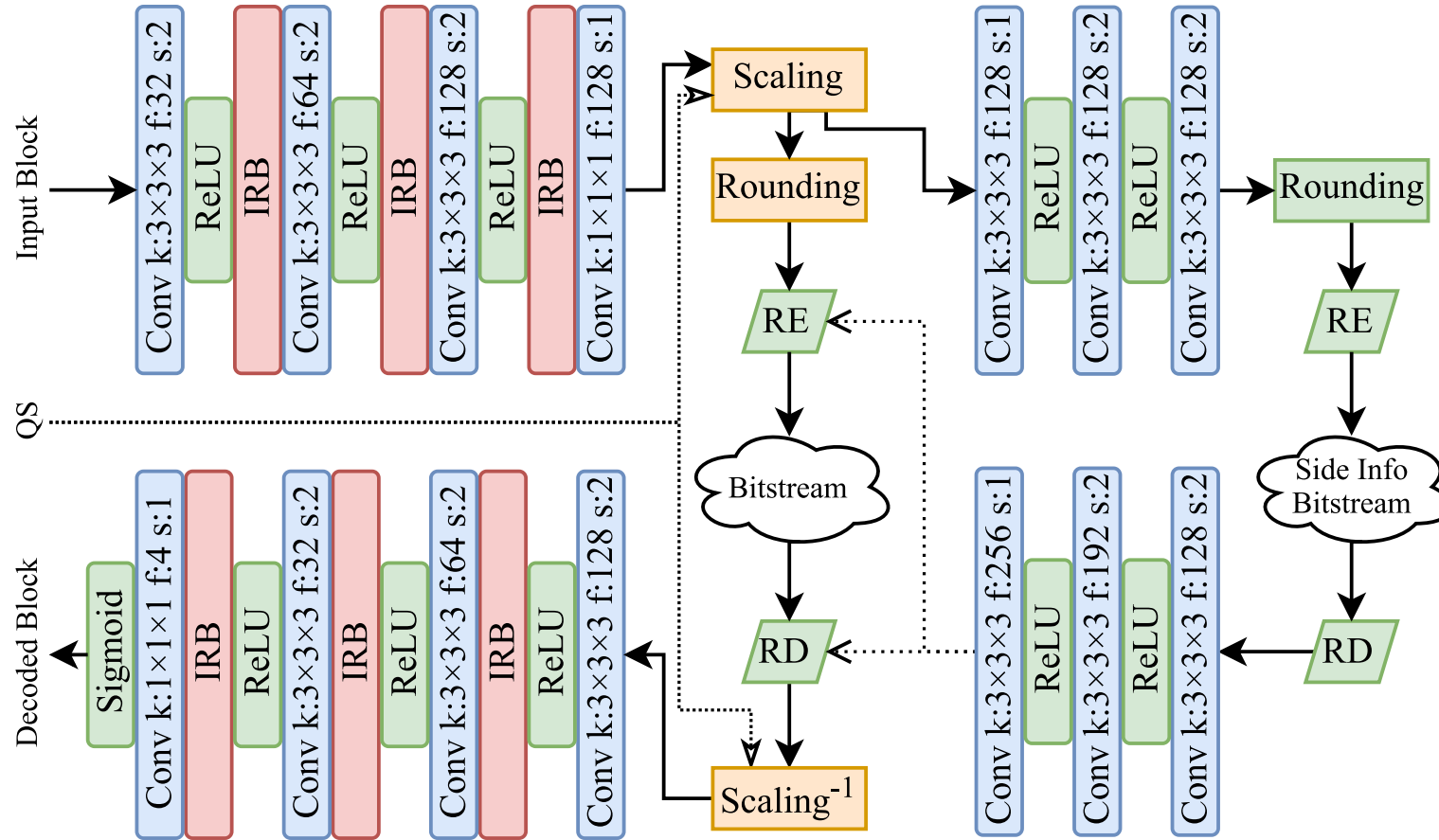
- Filter kernel slides across entire tensor, performing the computation for all voxels, occupied or empty

★ Sparse:

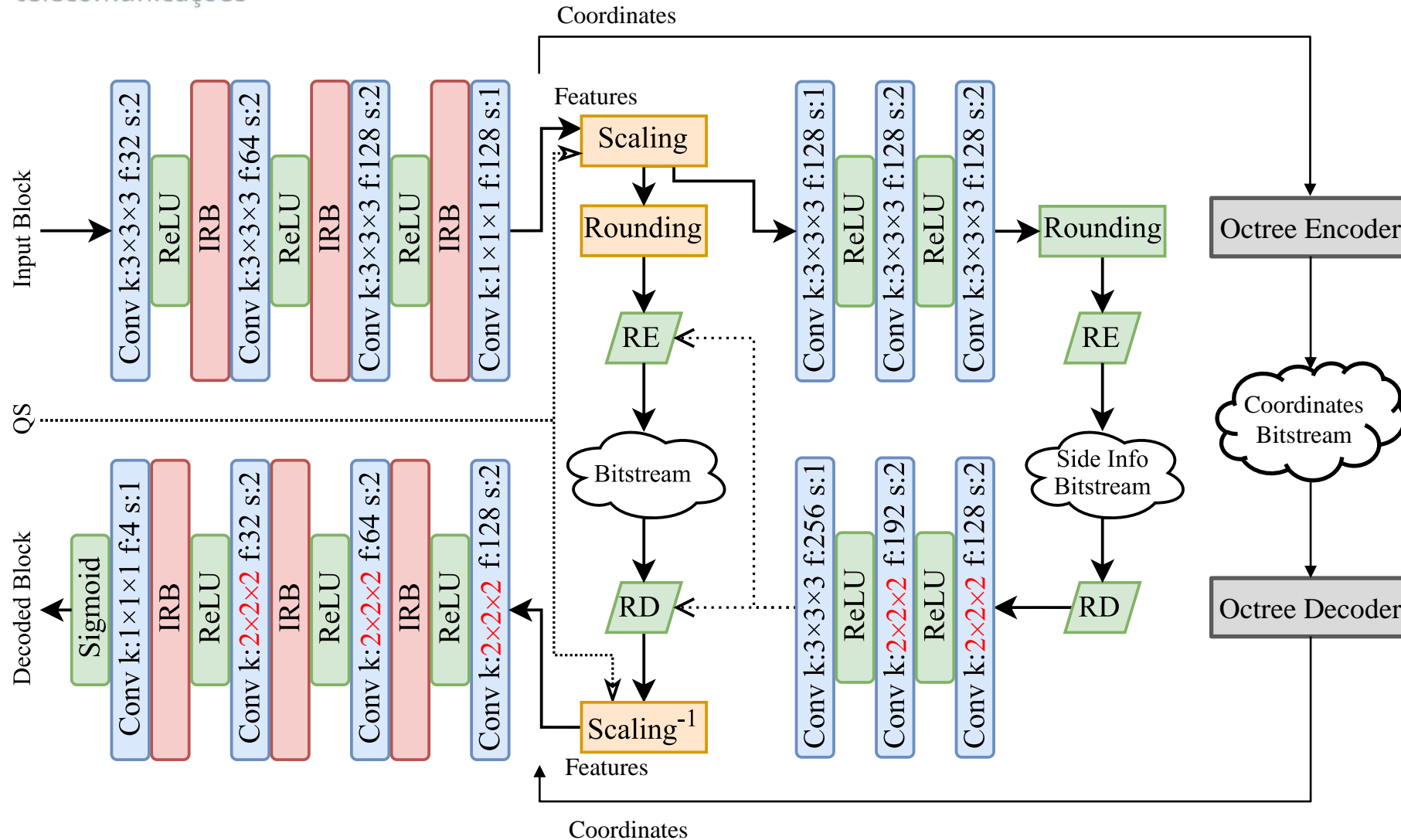
- Filter kernel is only applied on occupied voxels



Sparse Convolutions in the VM Coding Model



Sparse Convolutions in the VM Coding Model

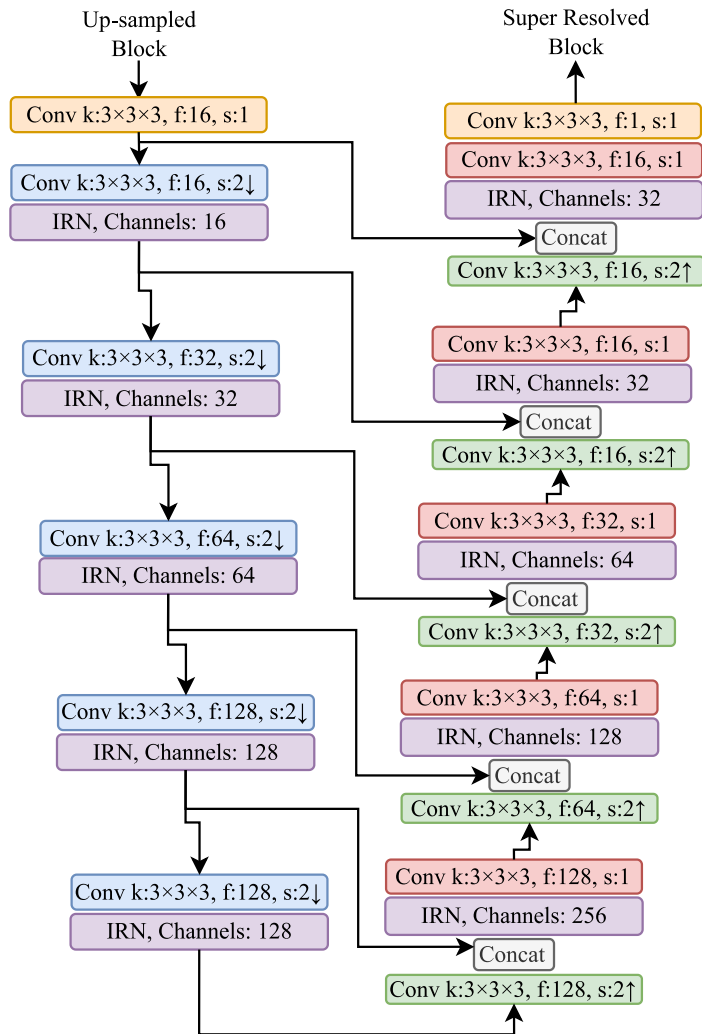


★ **Convolutions replaced with sparse equivalents**

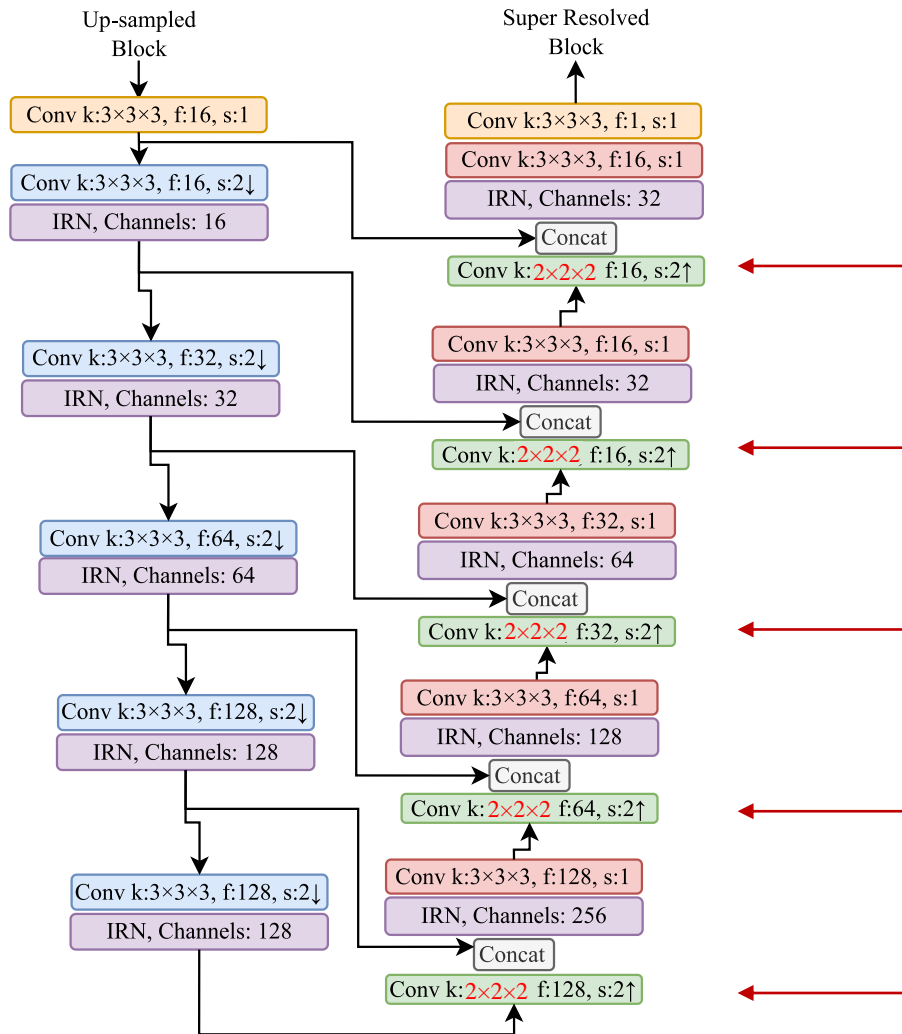
★ **Latents' coordinates losslessly coded with octree**

★ **Distortion loss changed from Focal Loss to Binary Cross Entropy**

Sparse Convolutions in the VM Super-Resolution Model



Sparse Convolutions in the VM Super-Resolution Model



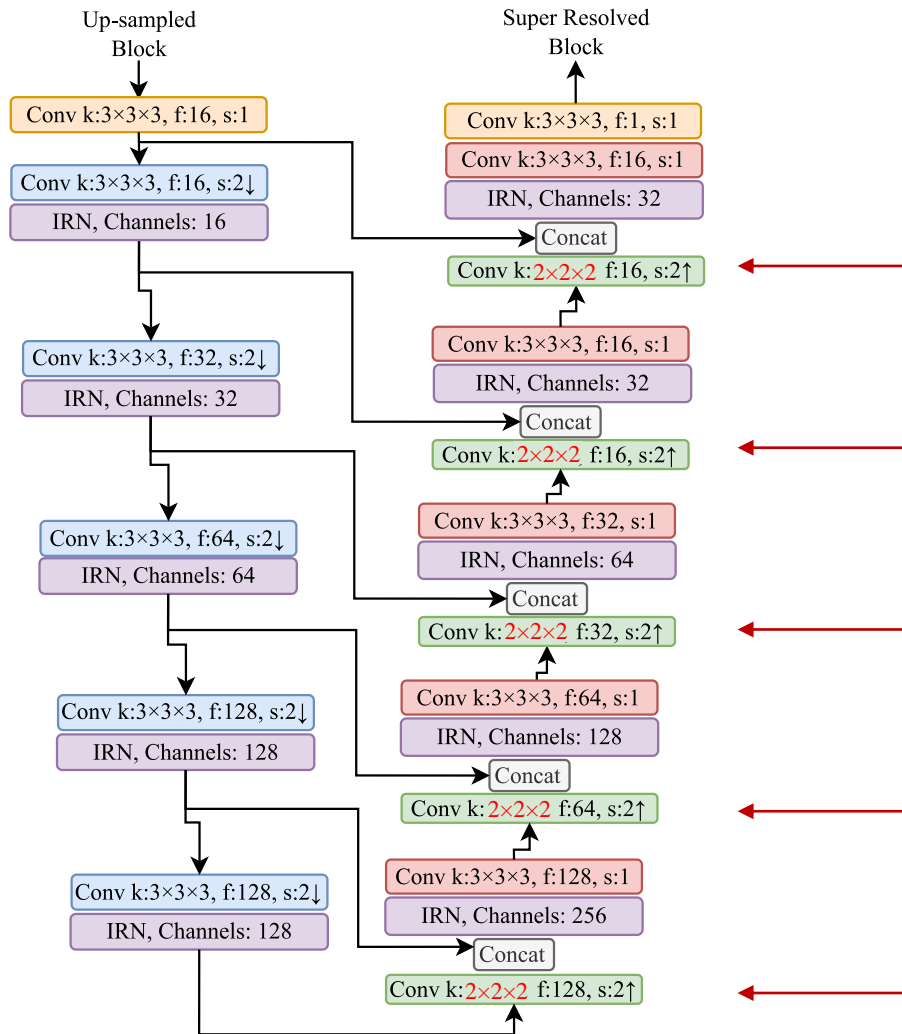
★ **Convolutions replaced with sparse equivalents**

★ **Distortion loss changed from Focal Loss to Binary Cross Entropy**

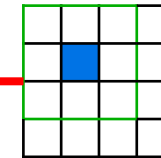
★ **All 5 up-sampling convolutional layers are generative**

- Consumes too much memory and takes too long (~ 1 hour each RD point in CPU)

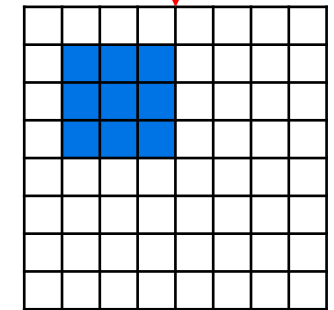
Sparse Convolutions in the VM Super-Resolution Model



Non-generative
up-sampling



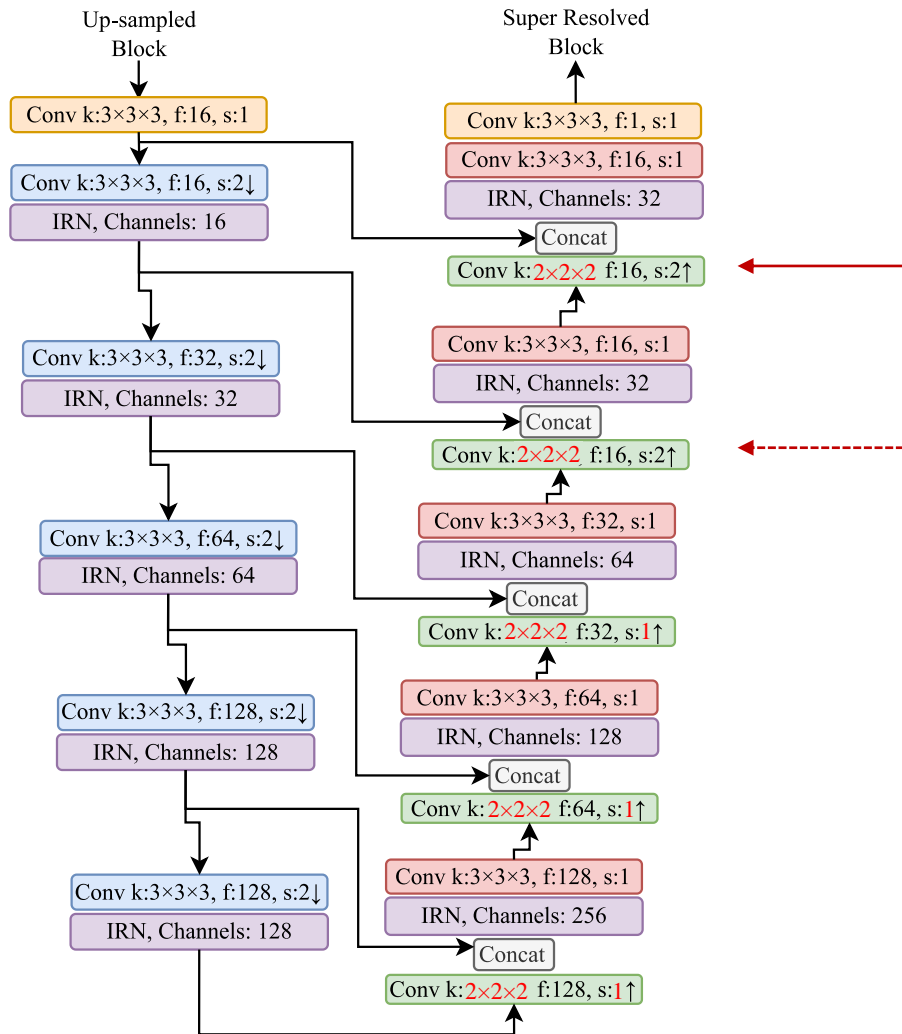
Generative up-
sampling



★ All 5 up-sampling convolutional layers are **generative**

- Consumes too much memory and takes too long (~ 1 hour each RD point in CPU)

Sparse Convolutions in the VM Super-Resolution Model



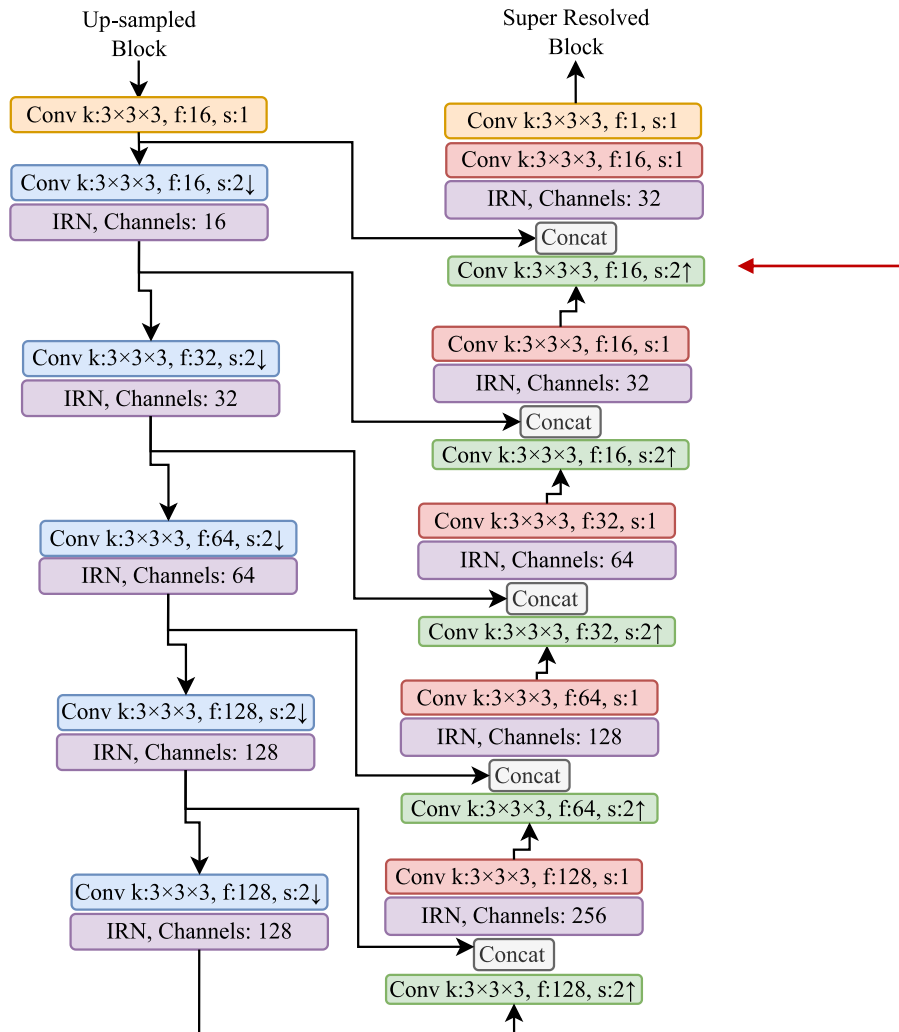
★ **Convolutions replaced with sparse equivalents**

★ **Distortion loss changed from Focal Loss to Binary Cross Entropy**

★ **Only the last (and second to last) up-sampling convolutional layer(s) is(are) generative, for SF=2(4)**

- Significant less memory and running time (~ 6 minutes each RD point in CPU)

Sparse Convolutions in the VM Super-Resolution Model



★ **Convolutions replaced with sparse equivalents**

★ **Distortion loss changed from Focal Loss to Binary Cross Entropy**

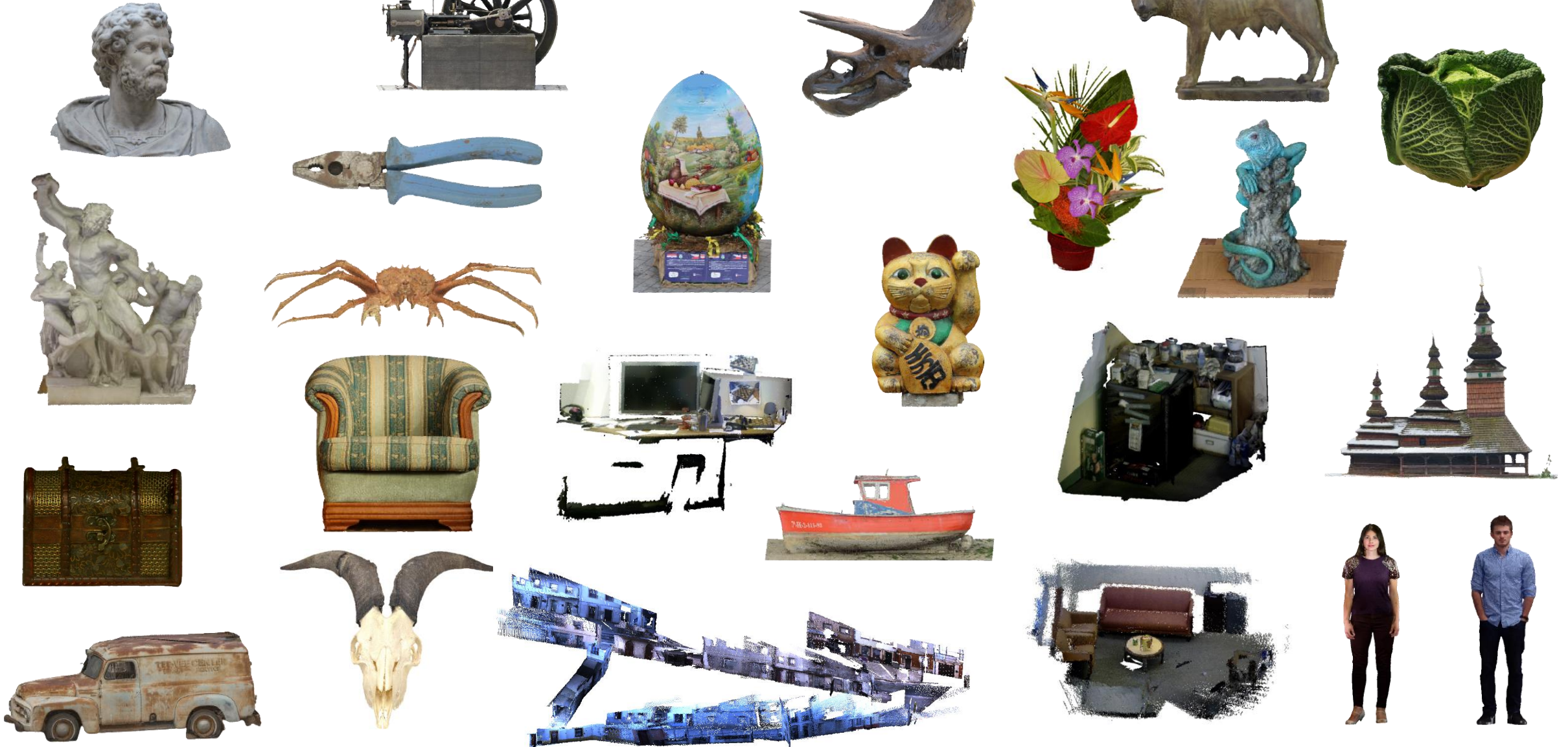
★ **Only the last up-sampling convolutional layer is generative, using $3 \times 3 \times 3$ for SF=2 and $5 \times 5 \times 5$ for SF=4**

- Significant less memory and running time (~ 5 minutes each RD point in CPU)



Performance Assessment: Sparse vs Dense Convolutions

Experimental Conditions: Test Material





instituto de
telecomunicações



Evaluating Super-Resolution Models

Experimental Conditions: Configurations

★ **Sparse coding + SR models VS dense coding + SR models**

★ **Block size = 128**

★ **Sampling factor = 2:**

- 5 generative layers – gen5
- 2 generative layers – gen2
- 1 generative layer – gen1
- 1 generative layer alternative – gen1alt

★ **Sampling factor = 4:**

- 5 generative layers – gen5
- 2 generative layers – gen2
- 1 generative layer – gen1
- 1 generative layer alternative – gen1alt

RD Performance: Bjontegaard-Delta (SF=2)

Reference: Dense VM with SR

PC	Sparse SR gen5				Sparse SR gen2				Sparse SR gen1				Sparse SR gen1alt			
	PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	-5.7%	0.22	-5.3%	0.21	-4.9%	0.18	-4.9%	0.18	-2.8%	0.09	-4.1%	0.16	-5.2%	0.20	-6.0%	0.23
Lagoon	-4.1%	0.17	-4.1%	0.19	-3.2%	0.13	-3.9%	0.18	-0.1%	-0.01	-2.6%	0.11	-2.7%	0.11	-2.9%	0.13
Van	-2.0%	0.07	-4.3%	0.17	0.9%	-0.04	-2.3%	0.08	4.6%	-0.19	0.3%	-0.04	0.3%	-0.02	-2.6%	0.09
StMichael	-4.0%	0.16	-6.9%	0.29	-0.6%	0.00	-5.7%	0.21	0.7%	-0.05	-4.7%	0.16	-3.5%	0.13	-8.3%	0.35
CowStatue	-31.8%	1.70	-33.6%	1.92	-26.4%	1.34	-29.9%	1.66	-24.8%	1.23	-28.2%	1.50	-26.6%	1.36	-29.7%	1.67
Bouquet	-2.3%	0.09	-4.7%	0.21	0.1%	-0.03	-2.1%	0.06	2.1%	-0.12	-0.1%	-0.04	-0.3%	-0.01	-2.9%	0.11
CatStatue	-5.7%	0.18	-3.7%	0.14	-3.6%	0.11	-1.7%	0.07	1.6%	-0.05	2.3%	-0.08	-4.3%	0.13	-2.5%	0.09
BodyScanBlueShirt	-4.6%	0.18	-5.6%	0.25	-5.1%	0.20	-7.1%	0.32	-0.4%	0.01	-1.6%	0.05	-4.3%	0.17	-5.6%	0.25
Cabbage	-11.5%	0.36	-7.9%	0.28	-8.7%	0.26	-7.8%	0.30	-6.0%	0.15	-8.3%	0.34	-10.3%	0.32	-6.6%	0.21
SteamEngine	-11.3%	0.38	-12.6%	0.41	-11.9%	0.40	-17.2%	0.57	-10.2%	0.33	-12.6%	0.40	-9.3%	0.29	-10.8%	0.31
ArmChair	-11.8%	0.31	-2.8%	0.06	-8.1%	0.21	-4.3%	0.11	-2.1%	0.04	-1.5%	0.03	-11.1%	0.29	-1.1%	-0.01
BoatJosefa	-3.0%	0.10	-8.2%	0.29	5.9%	-0.22	2.4%	-0.10	4.8%	-0.18	0.9%	-0.04	3.8%	-0.14	0.0%	-0.01
CapitolineWolf	-11.9%	0.40	-6.3%	0.22	-11.9%	0.41	-8.1%	0.29	-8.5%	0.25	-2.7%	0.07	-12.2%	0.42	-8.8%	0.32
KingCrab	-0.2%	0.01	-0.4%	0.01	-0.7%	0.03	-0.6%	0.02	-0.3%	0.01	-0.6%	0.02	-0.8%	0.03	-0.7%	0.02
WoodenChest	-38.3%	0.96	-40.8%	1.38	-36.5%	0.89	-39.7%	1.33	-28.5%	0.56	-31.3%	0.78	-29.4%	0.63	-31.5%	0.87
BodyScanOlia	-1.1%	0.03	-0.8%	0.02	0.0%	-0.01	-0.2%	0.00	1.1%	-0.05	0.9%	-0.06	0.0%	-0.01	0.0%	-0.02
PaintedEgg	-5.8%	0.26	-8.6%	0.41	-2.8%	0.12	-6.5%	0.30	0.6%	-0.03	-0.8%	0.02	-4.7%	0.21	-7.7%	0.35
Annibal	-16.1%	0.76	-15.6%	0.80	-16.2%	0.75	-15.5%	0.80	-14.8%	0.66	-14.3%	0.74	-16.2%	0.74	-15.7%	0.82
Iguana	-6.2%	0.31	-4.6%	0.23	-4.1%	0.19	-3.2%	0.15	-3.9%	0.19	-3.6%	0.18	-4.5%	0.22	-3.4%	0.15
Pliers	-7.1%	0.18	-4.6%	0.10	-6.5%	0.17	-8.4%	0.23	-10.0%	0.26	-13.3%	0.36	-9.8%	0.25	-9.7%	0.26
goat_skull	-6.5%	0.18	-1.3%	0.00	-4.3%	0.11	-0.1%	-0.04	-3.7%	0.08	-4.0%	0.15	-5.4%	0.15	-0.7%	-0.03
kinfudesk	-24.5%	0.91	-20.1%	0.95	-23.1%	0.86	-18.6%	0.90	-22.1%	0.82	-16.0%	0.73	-23.2%	0.86	-18.6%	0.91
kinfubooks	-15.5%	0.54	-12.7%	0.51	-13.8%	0.47	-11.5%	0.46	-12.0%	0.39	-11.8%	0.50	-13.4%	0.45	-11.3%	0.47
LivingRoom	-22.5%	0.48	-19.4%	0.52	-23.1%	0.51	-21.5%	0.60	-17.6%	0.22	-20.2%	0.54	-19.5%	0.41	-17.9%	0.48
RuaDeCoimbra	0.8%	-0.04	-6.6%	0.25	2.6%	-0.11	-4.8%	0.17	8.2%	-0.33	1.2%	-0.07	3.0%	-0.14	-8.0%	0.30
Average	-10.1%	0.36	-9.7%	0.39	-8.2%	0.28	-8.9%	0.35	-5.8%	0.17	-7.1%	0.26	-8.4%	0.28	-8.5%	0.33

RD Performance: Bjontegaard-Delta (SF=4)

Reference: Dense VM with SR

PC	Sparse SR gen5				Sparse SR gen2				Sparse SR gen1				Sparse SR gen1alt			
	PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2		PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	-15.1%	0.72	-16.5%	0.90	-8.4%	0.26	-9.5%	0.36	3.6%	-1.08	2.1%	-1.20	-13.9%	0.57	-14.6%	0.69
Lagoon	-19.6%	1.20	-19.4%	1.35	-16.1%	0.94	-16.2%	1.07	14.0%	-2.09	6.0%	-2.19	-16.8%	0.97	-16.5%	1.09
Van	-15.7%	0.85	-18.7%	1.16	-10.6%	0.54	-14.0%	0.80	23.1%	-2.45	12.8%	-2.44	-7.4%	0.32	-10.9%	0.58
StMichael	-22.3%	1.35	-24.9%	1.71	-12.5%	0.60	-15.7%	0.88	2.4%	-1.22	-1.6%	-1.24	-17.1%	0.96	-19.4%	1.24
CowStatue	-45.8%	2.26	-49.5%	2.46	-41.4%	1.73	-47.0%	2.07	3.9%	-1.70	2.2%	-1.59	-36.4%	1.26	-43.3%	1.61
Bouquet	-7.3%	0.19	-9.8%	0.36	-4.9%	0.01	-7.4%	0.17	3.7%	-0.78	2.0%	-0.80	-6.5%	0.11	-8.5%	0.25
CatStatue	-19.5%	1.03	-19.3%	1.19	-18.2%	0.89	-17.5%	1.00	18.3%	-3.91	29.3%	-4.16	-14.9%	0.67	-15.3%	0.81
BodyScanBlueShirt	-16.2%	0.90	-16.9%	1.10	-14.8%	0.80	-14.9%	0.93	31.7%	-3.02	26.9%	-3.35	-12.6%	0.66	-13.3%	0.81
Cabbage	-25.2%	1.46	-25.5%	1.82	-23.6%	1.31	-23.7%	1.63	1.5%	-2.42	-2.3%	-2.61	-21.3%	1.12	-21.4%	1.39
SteamEngine	-19.5%	0.83	-19.9%	0.84	-12.2%	0.34	-13.0%	0.30	5.0%	-1.36	3.8%	-1.82	-13.0%	0.40	-13.8%	0.35
ArmChair	-29.0%	1.34	-29.7%	1.65	-25.7%	1.09	-26.2%	1.33	33.2%	-3.86	39.0%	-3.58	-17.4%	0.50	-18.8%	0.67
BoatJosefa	-23.5%	1.14	-28.1%	1.43	-2.9%	0.08	-9.0%	0.32	6.6%	-0.73	4.7%	-0.74	-9.4%	0.36	-17.1%	0.72
CapitolineWolf	-26.6%	1.68	-26.1%	1.60	-17.2%	0.75	-15.6%	0.43	20.0%	-2.97	17.1%	-3.44	-15.5%	0.47	-15.4%	0.21
KingCrab	-10.2%	0.66	-10.1%	0.77	-8.3%	0.39	-8.1%	0.48	0.6%	-1.01	0.5%	-0.61	-8.7%	0.54	-8.7%	0.63
WoodenChest	-43.0%	1.36	-43.1%	1.48	-39.6%	1.06	-40.0%	1.15	9.8%	-2.34	15.8%	-2.64	-26.9%	0.19	-26.4%	0.09
BodyScanOlia	-10.6%	0.63	-12.2%	0.87	-7.0%	0.38	-10.4%	0.68	28.1%	-2.97	33.7%	-3.15	-6.2%	0.33	-9.4%	0.61
PaintedEgg	-19.6%	0.94	-20.8%	1.20	-11.3%	0.42	-14.7%	0.67	1.9%	-0.96	4.0%	-1.08	-8.8%	0.35	-13.4%	0.61
Annibal	-16.3%	0.82	-17.6%	1.11	-11.9%	0.44	-13.2%	0.73	-5.0%	-1.42	-2.6%	-1.65	-12.8%	0.54	-14.1%	0.84
Iguana	-13.1%	0.80	-12.7%	0.66	-13.8%	0.82	-13.4%	0.85	3.4%	-1.08	2.9%	-0.91	-13.4%	0.81	-11.8%	0.73
Pliers	-22.8%	1.41	-24.7%	1.59	-13.8%	0.44	-13.9%	0.41	12.7%	-2.99	23.0%	-2.85	-11.2%	0.28	-12.1%	0.27
goat_skull	-12.5%	0.48	-14.4%	0.73	-12.6%	0.45	-14.0%	0.67	-0.7%	-0.71	0.8%	-0.89	-12.7%	0.42	-14.3%	0.62
kinfudesk	-26.0%	1.42	-26.8%	1.74	-24.2%	1.30	-24.5%	1.54	-15.9%	0.50	-12.7%	0.38	-19.6%	1.14	-24.4%	1.53
kinfubooks	9.4%	-0.13	-2.9%	0.07	-9.3%	0.49	-6.4%	0.37	-13.2%	0.48	10.7%	-0.82	1.6%	-0.23	-3.0%	0.17
LivingRoom	-48.2%	0.44	-17.1%	0.55	-54.1%	0.73	-29.0%	0.98	-47.4%	0.28	-18.9%	0.44	-50.1%	0.51	-22.1%	0.78
RuaDeCoimbra	-6.5%	0.28	-9.1%	0.37	-2.8%	0.02	-6.8%	0.29	3.9%	-1.38	0.8%	-1.32	-5.7%	0.16	-7.6%	0.30
Average	-20.2%	0.96	-20.6%	1.15	-16.7%	0.65	-17.0%	0.80	5.8%	-1.65	8.0%	-1.77	-15.1%	0.54	-15.8%	0.70

Observations

- ★ **Using 5 generative up-sampling convolutional layers provides the best overall results; however, this approach is very intensive computationally, and likely will not allow processing the full PC at once**
- ★ **Using 1 generative up-sampling convolutional layer provides the worst results, especially for SF=4 where it fails to outperform the dense VM**
- ★ **The alternative implementation using 1 generative up-sampling convolutional layer provides similar results as when using 2 generative layers, with slightly lower computational complexity (coding time and memory)**



instituto de
telecomunicações



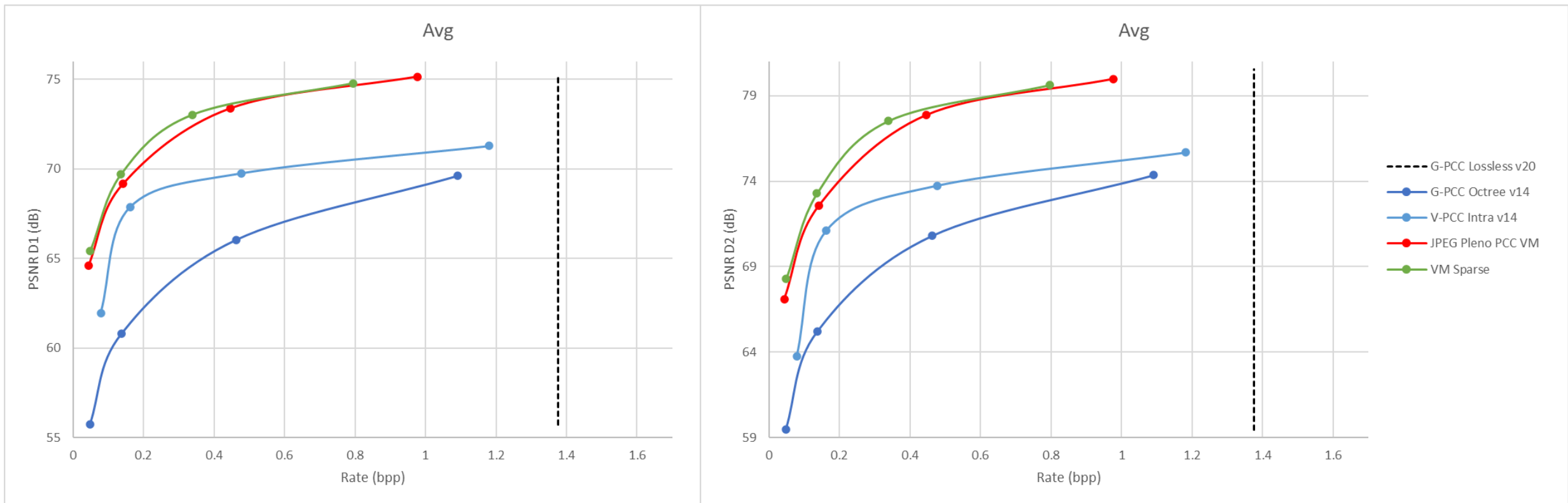
Evaluating Overall Codec on JPEG CTTC

JPEG Common Training and Testing Conditions

★ Target rates for geometry-only coding: 0.05, 0.15, 0.5, 1.5 (50% of total rate)

Point cloud	R4				R3				R2				R1			
	λ	BS	SF	SR	λ	BS	SF	SR	λ	BS	SF	SR	λ	BS	SF	SR
DinoSkull	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Lakoon	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Van	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
StMichael	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
CowStatue	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Bouquet	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
CatStatue	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
BodyScanBlueShirt	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Cabbage	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
SteamEngine	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
ArmChair	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
BoatJosefa	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
CapitolineWolf	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
KingCrab	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
WoodenChest	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
BodyScanOlia	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
PaintedEgg	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Annibal	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Iguana	0.0025	128	1	0	0.01	128	1	0	0.005	128	2	1	0.025	128	2	1
Pliers	0.0025	128	1	0	0.005	128	1	0	0.025	128	1	0	0.025	128	2	1
goat_skull	0.0025	128	1	0	0.01	128	1	0	0.01	128	2	1	0.01	128	4	1
kinfudesk	0.005	128	1	0	0.01	128	2	1	0.01	128	4	1	0.05	128	4	0
kinfubooks	0.0025	128	2	1	0.01	128	2	1	0.05	128	2	0	0.05	128	4	0
LivingRoom	0.0025	128	1	0	0.01	128	1	0	0.01	128	2	1	0.01	128	4	1
RuaDeCoimbra	0.0025	128	1	0	0.01	128	1	0	0.01	128	2	1	0.01	128	4	1

Average RD Performance



RD Performance: Bjontegaard-Delta

Reference: JPEG Pleno PCC VM

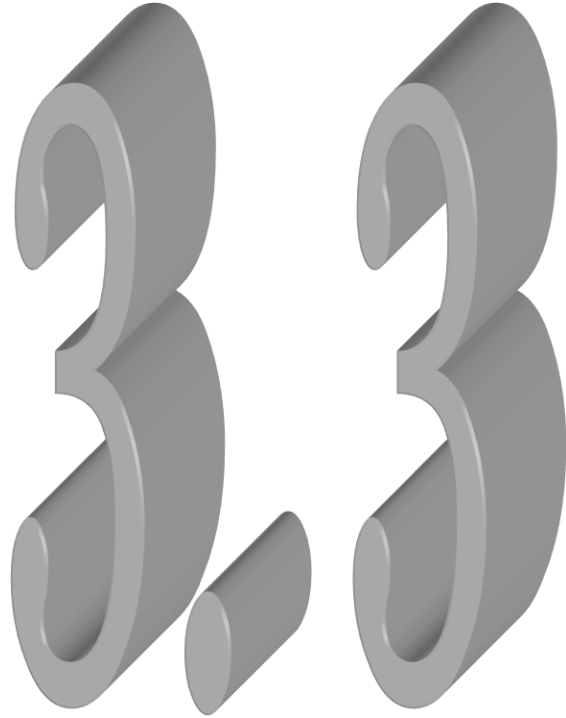
PC	VM Sparse			
	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	-10.9%	0.43	-12.5%	0.60
Lagoon	-10.8%	0.42	-11.3%	0.53
Van	-13.6%	0.56	-14.9%	0.73
StMichael	-12.6%	0.49	-16.4%	0.74
CowStatue	-20.8%	0.97	-23.6%	1.35
Bouquet	-9.3%	0.36	-11.8%	0.53
CatStatue	-19.9%	0.63	-20.4%	0.81
BodyScanBlueShirt	-21.9%	1.03	-22.9%	1.29
Cabbage	-11.6%	0.37	-5.8%	0.32
SteamEngine	-36.3%	1.60	-43.6%	2.29
ArmChair	-21.8%	0.57	-14.5%	0.45
BoatJosefa	4.5%	-0.15	2.0%	-0.07
CapitolineWolf	-25.1%	1.04	-31.6%	1.66
KingCrab	-4.5%	0.22	-4.5%	0.26
WoodenChest	-27.6%	0.79	-26.4%	1.14
BodyScanOlia	-24.9%	1.19	-26.5%	1.51
PaintedEgg	-20.7%	0.89	-22.8%	1.10
Annibal	-12.3%	0.53	-8.9%	0.48
Iguana	-7.9%	0.41	-6.0%	0.34
Pliers	-7.2%	0.31	-6.0%	0.31
goat_skull	-15.8%	0.35	-10.8%	0.43
kinfudesk	-18.9%	0.85	-24.8%	1.59
kinfubooks	-1.2%	0.05	-0.7%	0.05
LivingRoom	-35.7%	0.85	-30.6%	0.86
RuaDeCoimbra	4.2%	-0.13	-8.7%	0.36
Average	-15.3%	0.58	-16.2%	0.79

Observations

- ★ **Adopting a sparse tensor representation in the JPEG Pleno PCC VM provides significant RD performance improvements**

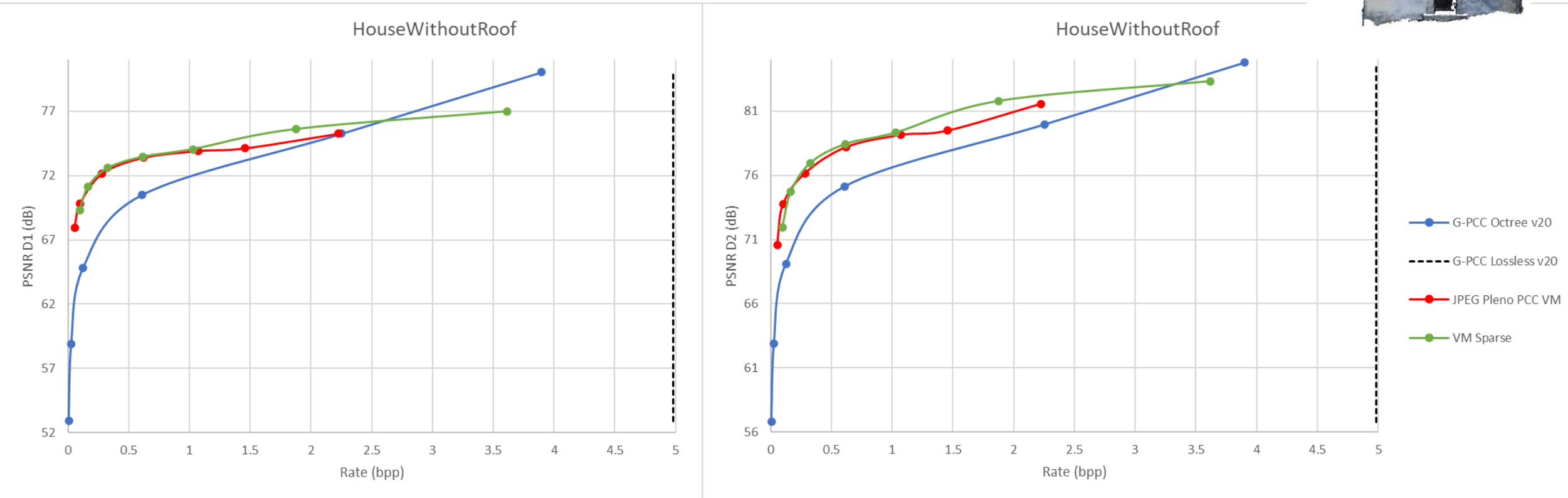


instituto de
telecomunicações



Evaluating Overall Codec on Sparse MPEG PCs

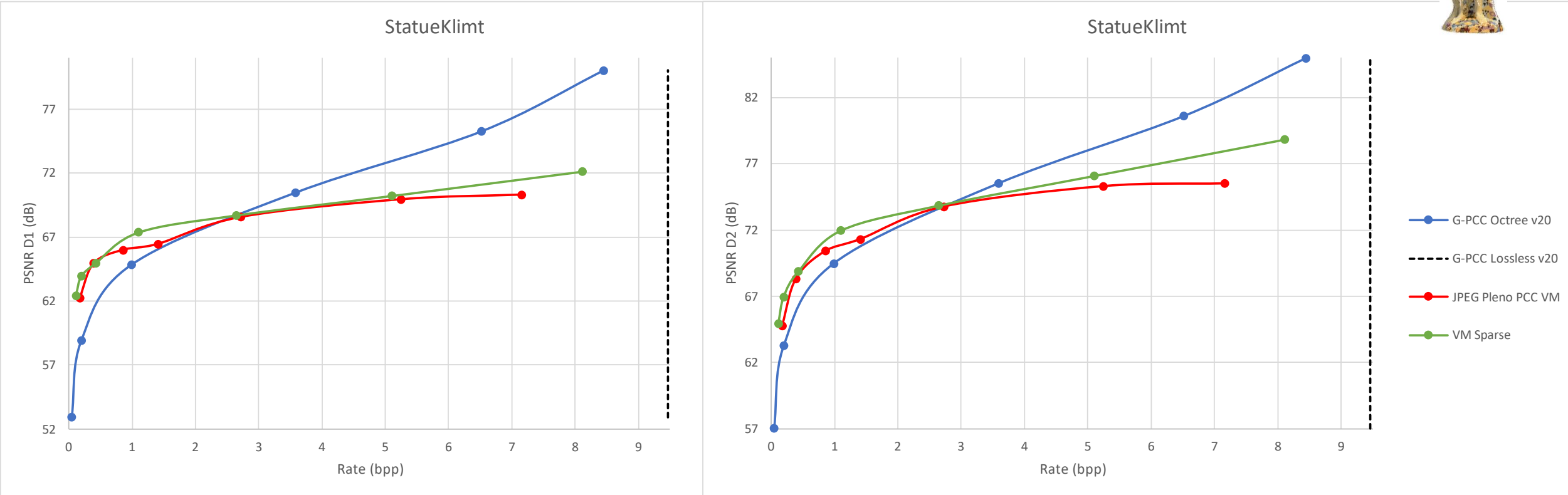
RD Performance: *House Without Roof*



PC	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
House Without Roof	-9.3%	0.22	-3.5%	0.22

Sparse vs Dense VM

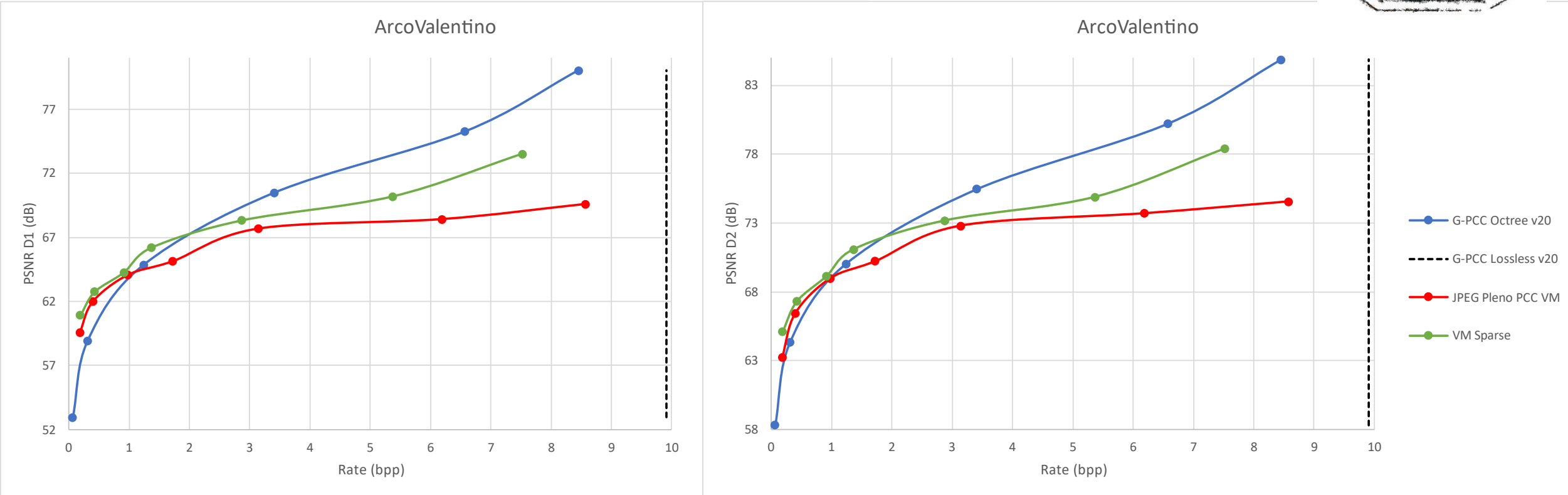
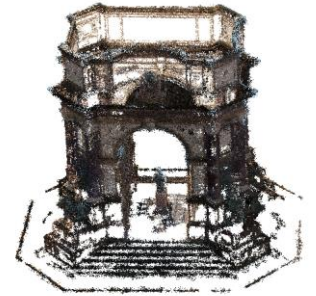
RD Performance: *Statue Klimt*



PC	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
Statue Klimt	-22.8%	0.55	-21.5%	0.74

Sparse vs Dense VM

RD Performance: *Arco Valentino*



PC	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
Arco Valentino	-31.9%	1.15	-26.6%	1.04

Sparse vs Dense VM

Observations

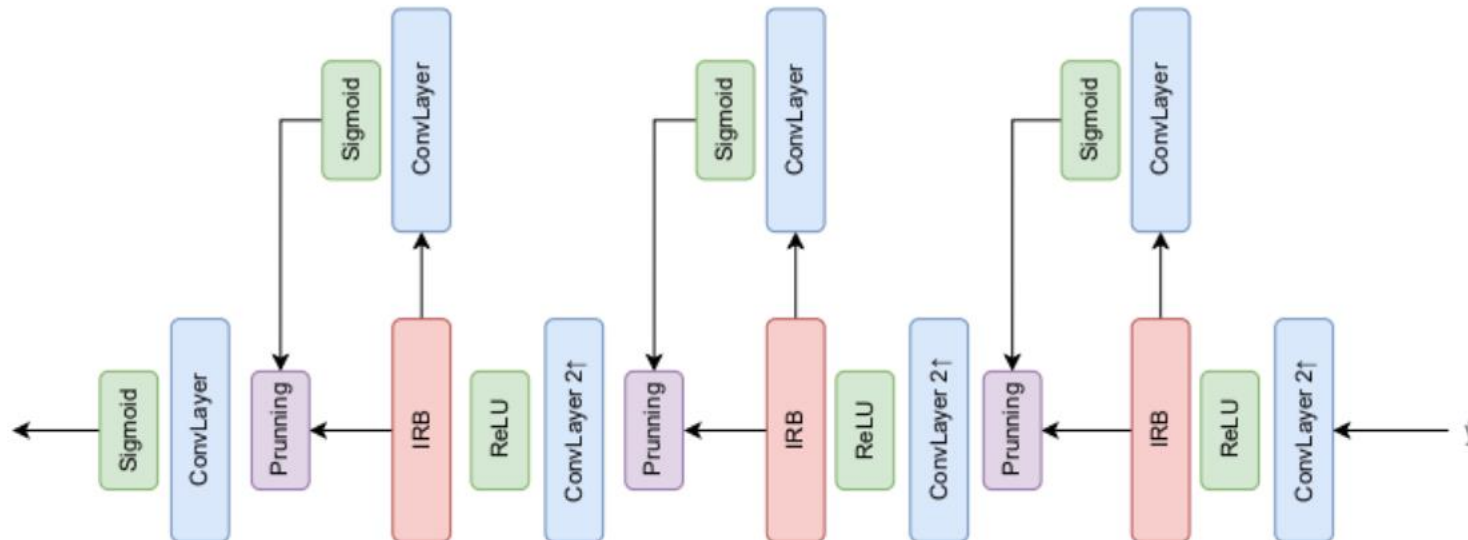
- ★ **The gains provided by adopting a sparse tensor representation in the JPEG Pleno PCC VM are even more significant for sparse PC content**



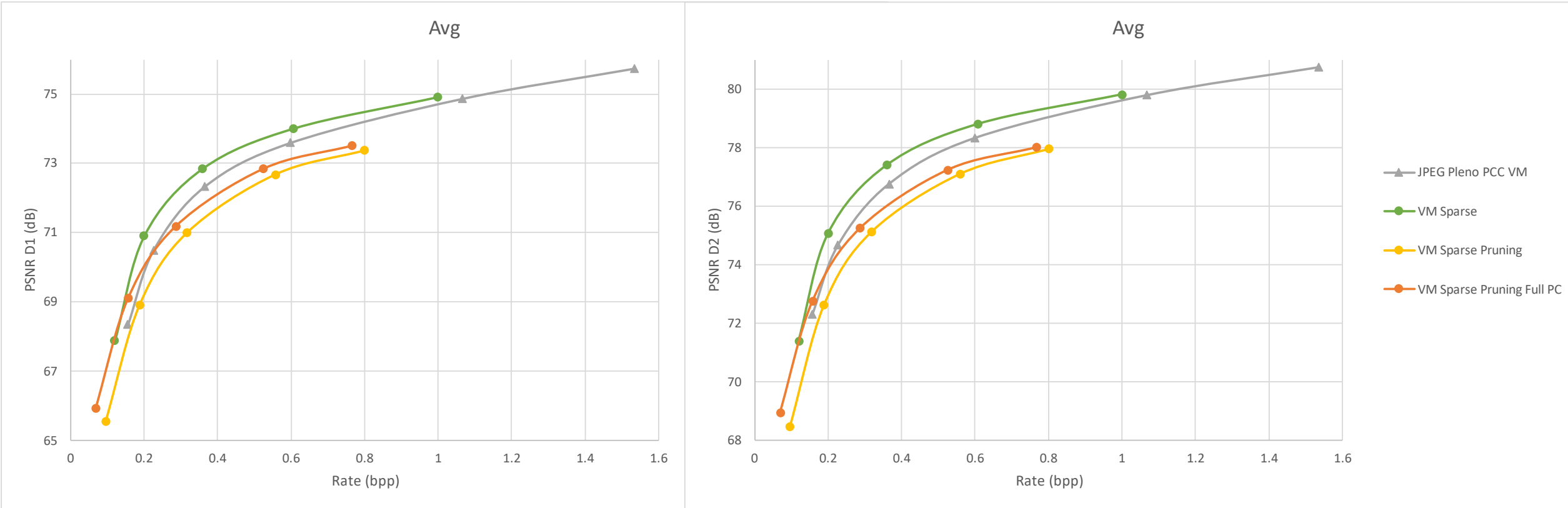
Sparse Convolutions in the VM: Pruning

Pruning

- ★ **With sparse convolutions, generative up-sampling layers create many new points (each occupied voxel leads to a number of voxels = size of the kernel), significantly increasing the memory requirements during coding**
- ★ **A pruning operation after each up-sampling layer reduces the number of points, and thus the memory usage, allowing to code the full point cloud at once**



Average RD Performance



RD Performance: Bjontegaard-Delta

Reference: VM Sparse

PC	VM Sparse Pruning			
	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	14.0%	-0.60	18.0%	-0.86
Lagoon	18.1%	-0.70	17.8%	-0.83
Van	15.5%	-0.70	21.1%	-1.00
StMichael	10.5%	-0.47	14.3%	-0.70
CowStatue	23.8%	-0.87	29.1%	-1.22
Bouquet	10.0%	-0.49	16.1%	-0.76
CatStatue	31.6%	-0.87	31.4%	-1.07
BodyScanBlueShirt	19.9%	-0.92	19.2%	-1.07
Cabbage	32.1%	-0.61	33.0%	-0.75
SteamEngine	10.9%	-0.44	21.7%	-0.82
ArmChair	29.8%	-0.73	30.7%	-0.91
BoatJosefa	10.6%	-0.62	13.4%	-0.79
CapitolineWolf	24.9%	-0.89	28.4%	-1.16
KingCrab	15.5%	-0.72	17.8%	-0.94
WoodenChest	24.7%	-0.49	31.6%	-0.96
BodyScanOlia	23.8%	-1.02	23.1%	-1.19
PaintedEgg	15.6%	-0.64	20.4%	-0.85
Annibal	13.4%	-0.54	17.0%	-0.78
Iguana	13.5%	-0.71	13.8%	-0.81
Pliers	19.4%	-0.79	18.2%	-0.94
goat_skull	1.3%	-0.09	8.0%	-0.52
kinfudesk	43.9%	-1.00	33.3%	-1.18
kinfubooks	102.2%	-2.58	93.1%	-2.62
LivingRoom	12.9%	-0.19	21.0%	-0.36
RuaDeCoimbra	-6.7%	0.15	-0.7%	-0.24
Average	21.2%	-0.70	23.6%	-0.93

PC	VM Sparse Pruning Full PC			
	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	-2.9%	-0.02	2.5%	-0.29
Lagoon	-2.0%	-0.05	-0.8%	-0.12
Van	-3.5%	-0.03	3.3%	-0.29
StMichael	-5.4%	-0.01	2.5%	-0.32
CowStatue	6.6%	-0.31	13.7%	-0.61
Bouquet	-8.6%	-0.02	-0.8%	-0.29
CatStatue	9.4%	-0.29	9.5%	-0.38
BodyScanBlueShirt	2.7%	-0.27	0.9%	-0.23
Cabbage	3.6%	-0.11	8.8%	-0.24
SteamEngine	-2.9%	-0.01	8.2%	-0.44
ArmChair	3.7%	-0.16	8.0%	-0.30
BoatJosefa	-1.1%	-0.19	1.3%	-0.31
CapitolineWolf	5.9%	-0.25	7.9%	-0.37
KingCrab	-1.9%	-0.07	1.8%	-0.26
WoodenChest	-11.2%	0.26	32.3%	-0.90
BodyScanOlia	4.5%	-0.27	2.5%	-0.24
PaintedEgg	-2.0%	-0.14	9.0%	-0.49
Annibal	1.2%	-0.18	4.4%	-0.32
Iguana	-1.8%	-0.12	1.9%	-0.25
Pliers	7.6%	-0.38	6.0%	-0.43
goat_skull	-19.7%	0.25	-13.5%	0.07
kinfudesk	29.3%	-0.73	12.5%	-0.64
kinfubooks	104.8%	-2.93	94.7%	-2.78
LivingRoom	-28.5%	0.51	-16.0%	0.22
RuaDeCoimbra	-28.5%	0.64	-21.5%	0.44
Average	2.4%	-0.20	7.2%	-0.39

Reference: JPEG Pleno PCC VM

PC	VM Sparse Pruning Full PC			
	PSNR-D1		PSNR-D2	
	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR
DinoSkull	-8.6%	0.21	-2.6%	-0.06
Lagoon	-4.5%	0.06	-2.3%	-0.01
Van	-7.3%	0.18	-4.6%	0.11
StMichael	-5.6%	0.02	1.1%	-0.20
CowStatue	-5.9%	0.14	0.2%	-0.06
Bouquet	-11.8%	-0.06	-6.0%	-0.28
CatStatue	-5.9%	0.11	-6.1%	0.12
BodyScanBlueShirt	-1.2%	0.02	-1.9%	0.06
Cabbage	-6.2%	0.05	2.5%	-0.08
SteamEngine	-8.4%	0.21	-1.4%	-0.03
ArmChair	-4.2%	0.05	-0.1%	-0.03
BoatJosefa	3.2%	-0.27	3.3%	-0.33
CapitolineWolf	-3.5%	0.09	-4.0%	0.11
KingCrab	-7.6%	0.22	-3.2%	0.05
WoodenChest	-30.4%	0.75	-3.4%	0.02
BodyScanOlia	-4.0%	0.09	-5.0%	0.14
PaintedEgg	-6.3%	0.04	1.2%	-0.27
Annibal	-4.2%	0.03	-2.6%	-0.03
Iguana	-3.4%	0.02	0.7%	-0.11
Pliers	5.9%	-0.23	5.1%	-0.26
goat_skull	-15.9%	0.15	-10.2%	-0.01
kinfudesk	-31.1%	0.83	-19.8%	0.37
kinfubooks	16.0%	-0.90	14.5%	-1.02
LivingRoom	-41.5%	1.24	-33.7%	0.67
RuaDeCoimbra	-41.6%	0.97	-37.5%	1.02
Average	-9.3%	0.16	-4.6%	0.00

Observations

- ★ **When using pruning, the overall RD performance for the adoption of a sparse tensor representation in the JPEG Pleno PCC VM is reduced**
- ★ **Despite showing some gains at low bitrates, the quality seems to saturate faster at higher rates, failing to outperform even the dense VM**
- ★ **The advantages of pruning are related to the reduced computational complexity, with both training and coding time being significantly reduced**
- ★ **Furthermore, with the reduced memory requirements, it is possible to code the full 10-bit PCs at once, providing significant RD performance improvements (though still below the performance without pruning)**



instituto de
telecomunicações



Conclusions

Conclusions

- ★ **Sparse convolutions provide a significant advantage over the dense convolutions, both in terms of RD performance and coding time, especially for sparser PC content**
- ★ **Pruning allows a considerable reduction of the computational complexity, both in terms of time and memory, being able to code the full PC at once; however, the RD performance is negatively impacted**

Proposal

The IT/IST/IPLeiria team proposes the adoption of the sparse tensor representation, and a continuation of the study on pruning