

MSPR 2x: 多重采样相位重建

面向实时渲染的 Tile-Local 空间上采样算法

技术报告

摘要

本报告介绍 MSPR 2x (Multisample Phase Reconstruction), 一种面向实时渲染的确定性相位重建算法。它从每个低分辨率像素的四个 resolve 前 MSAA 颜色 sample 重建四个输出相位, 面向 tile-based GPU 在渲染 pass 内直接生成 2x 结果, 并避免持久化全屏中间纹理。

历史九场景矩阵以 Direct 2x 为参考, 对比线性插值、FSR1 EASU+RCAS、MetalFX Spatial、MSPR-Quality 和 MSPR-Lite; 画质与 GPU 时间采用同一场景集合和执行契约。MSPR-Quality 以 30.301 dB 取得最高均值, 完整路径 GPU 均值为 3.656 ms, 比 Bilinear 高 1.067 ms; MSPR-Lite 达到 30.065 dB 和 3.218 ms, 仅比 Bilinear 高 0.629 ms。

证据边界。九场景矩阵来自旧版在同一台 iPhone 13 (iPhone14,5, iOS 18.2.1) 上的捕获, 并使用同帧 Direct 2x 参考。**历史资产声明:** Temple 与全屏图片两行使用的资产来自 Apple [可下载的 order-independent-transparency Demo \[1\]](#), 且仅用于测试。考虑到许可证和再分发边界, 本项目不发布这些资产、对应运行时场景或原始日志; 用户可从所链接 Apple 官方页面自行下载, 并遵守 Apple 的适用条款。这两行及其对历史画质与完整路径性能均值的贡献, 仅作为不可由当前发布包复现的证据保留; 当前源码复现覆盖七个项目原创程序化场景。更广的 100 帧完整渲染场景集保持原有校准离线 sample 协议, 不是 Metal imageblock 硬件 dump。

目录

1 结论摘要	2
2 算法与渲染契约	2
2.1 为什么 MSPR 要在 resolve 前执行	2
2.2 MSPR-Quality 参数标定	3
2.3 在实际渲染场景中调参	3
3 评测协议	3
3.1 历史九场景方法矩阵	3
3.2 更广的完整渲染场景集	4
3.3 指标与输入契约	4
4 画质结果	5
4.1 历史九场景空间方案比较	5
4.2 当前发布版典型 case	6
4.3 更广的 100 帧完整渲染场景集	9
4.4 完整实时渲染场景结果	10
5 性能与带宽证据	11
5.1 设备与测量方法	11
5.2 同一矩阵上的完整路径对比	12
6 与主流移动上采样器的关系	12

1 结论摘要

1. **MSPR 直接利用 resolve 前的子像素证据。**它用固定相位系数和受限内容自适应混合，从四个 MSAA sample 与局部 resolved color 重建四个输出相位，尤其适合几何边缘、透明边界和高光轮廓。
2. **两种 MSPR 配置都以较小的完整路径增量获得更高画质。**MSPR-Quality 和 MSPR-Lite 的历史九场景均值均高于 MetalFX、线性插值和 FSR1。相对同一矩阵上 2.589 ms 的 Bilinear 完整路径均值，MSPR-Quality 增加 1.067 ms、提升 1.021 dB；MSPR-Lite 增加 0.629 ms、提升 0.785 dB。
3. **MSPR-Quality 是画质优先工作点。**历史九场景平均 PSNR 为 30.301 dB，比 MetalFX、线性插值和 FSR1 分别高 0.891、1.021 和 2.934 dB；100 帧完整渲染场景中也比 Bicubic 高 0.815 dB。
4. **MSPR-Lite 是成本优先工作点。**它的历史九场景均值为 30.065 dB，仅比 MSPR-Quality 低 0.237 dB；移除 threadgroup staging 与 barrier 后，完整路径均值为 3.218 ms，在同一负载契约下比 Bilinear 高 0.629 ms。
5. **两种部署配置覆盖不同预算。**MSPR-Quality 使用 3×3 resolved 邻域来增强内容自适应能力；MSPR-Lite 使用当前像素四个 sample 和左右两个 resolved tap，以更简单的固定六 tap 路径换取最低成本。
6. **发布包提供当前复现入口。**物理 iPhone 上的一条命令可重新生成七个已分发程序化场景的 PSNR、代表性案例原始图和报告版式对比图；同机运行时 MetalFX、MSPR、FSR1、线性插值与 Direct 2× 使用相同帧和颜色契约。

2 算法与渲染契约

2.1 为什么 MSPR 要在 resolve 前执行

FSR1 是 resolved-color 空间上采样器：EASU 做方向自适应边缘重建，RCAS 负责锐化 [2, 3]。SGSR1 同样只读取一张 resolved texture，在一个面向移动 GPU 的 shader 中组合 12-tap 类 Lanczos 滤波和自适应锐化 [4]。MetalFX Spatial 读取输入颜色纹理，通过 Apple runtime 输出更大纹理 [5]。三者通常都从传统 resolve 之后开始。

MSPR 则在四个 MSAA 颜色仍处于 tile-local 状态时读取它们。对于输出相位 $p \in \{0, 1, 2, 3\}$ ：

$$L_p = \sum_{s=0}^3 w_{p,s} C_s, \quad (1)$$

$$B_p = 0.5625R + 0.1875R_x + 0.1875R_y + 0.0625R_{xy}, \quad (2)$$

$$\tilde{B}_p = \text{clip}(B_p + \gamma D, 0, 1), \quad (3)$$

$$O_p = \text{clip}(\tilde{B}_p + \alpha (L_p - \tilde{B}_p), 0, 1), \quad (4)$$

其中 C_s 是 resolve 前 sample， R 是 resolve， R_x 、 R_y 和 R_{xy} 是按输出相位选择的邻居， $D = R - 0.25(R_L + R_R + R_T + R_B)$ 是局部 detail。 γ 根据 resolved contrast 限制 detail 补偿， α 由 resolved contrast 和 sample disagreement 控制。四个相位分别写入对应的 2× 输出像素。

MSPR-Lite 保留相同的四相位输出，但将内容自适应邻域替换为固定六 tap 重建：

$$O_p^{\text{Lite}} = \text{clip}\left(\sum_{s=0}^3 a_{p,s} C_s + b_{p,L} R_L + b_{p,R} R_R, 0, 1\right), \quad (5)$$

其中 R_L 和 R_R 是从 imageblock 直接读取的左右 resolved color；当邻居不是完整四 sample coverage 时回退到当前像素 resolve。该路径不建立 3×3 threadgroup 邻域，也不需要 barrier。固定系数位于 `formulaV2MSAATile`，因此 Lite 与 Quality 都在同一个渲染 pass 内直接写出最终 2× 图像。

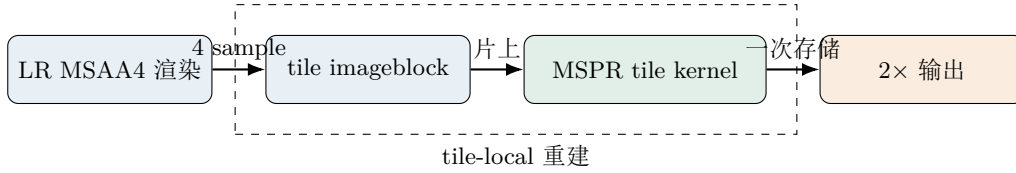


图 1: 目标 Metal 执行方式。四 sample 数据不会被物化为一张全屏外存纹理。

2.2 MSPR-Quality 参数标定

发布版画质配置使用下面这组固定相位系数、contrast blend、受限 detail 和 sample-disagreement gate。当前 release-shader 场景矩阵用于在多个实验配置之间选择默认参数，因此该矩阵属于发布验证证据，而不是未参与选择的独立 benchmark。另一组 100 帧完整渲染场景没有用于拟合这些系数。

$$W_{\text{MSPR-Quality}} = \begin{bmatrix} 0.63305 & 0.11351 & 0.27293 & -0.01922 \\ 0.39027 & 0.56770 & 0.02330 & 0.01899 \\ 0.01989 & 0.02368 & 0.56413 & 0.39257 \\ -0.01760 & 0.27210 & 0.11202 & 0.63377 \end{bmatrix}. \quad (6)$$

contrast 区间为 0.0400–0.2000，低/高 blend weight 为 0.7000/0.3000，受限 detail gain 为 0.3000，sample disagreement floor 为 0.7000。每一行都约束为单位 DC gain。机器可读配置位于 Reports/MSPR/parameters/mspr-quality.json，shader 中使用相同的 half 精度常量。

2.3 在实际渲染场景中调参

调参必须先固定输入契约：MSAA sample 顺序与位置、tone map、颜色空间、attachment 格式、曝光和 Direct 2x 参考都不能在搜索过程中变化。建议准备按内容族分层的开发集，至少覆盖几何斜边、alpha cutout、透明混合、高光、细线、重复纹理、粒子、UI 和低对比渐变；每个内容族同时保留静态帧和相机运动序列。

1. **拟合相位矩阵**。对四个输出相位分别以 $C_0 \dots C_3$ 拟合 Direct 2x target，使用带正则项的约束最小二乘，并强制每行系数和为 1。先只优化相位矩阵，避免 gate 同时变化掩盖系数问题。
2. **标定 contrast gate**。从开发集 luminance gradient 的分位数初始化 contrast_low/high。提高高对比区域的 Local 权重会增强相位边缘，同时也会放大错误 sample 顺序、纹理振铃和 HDR 过冲。
3. **标定 sample-disagreement gate**。用四 sample 的最大通道差分布确定 sample_low/high；在真实几何边缘和透明边界上提高 sample_floor，在噪声纹理或不稳定高光上降低它。
4. **最后调整 detail**。detail_gain 只用于补偿局部基线的软化。应从 0 开始增加，并同时监控 Scharr-gradient L1、负 lobe 过冲和时间序列闪烁，不能只优化单帧 PSNR。
5. **按内容族验收**。目标函数同时使用平均 PSNR、P10 PSNR、L1、gradient L1 和各内容族最差值；选定参数后冻结 JSON 与 SHA-256，再在未参与搜索的完整渲染序列上运行一次验证。

参数只改变常量，不改变 sample/邻域读取、barrier 和写出拓扑，因此同一 MSPR-Quality kernel 内的参数版本通常具有相同 GPU 成本。若目标项目的 sample pattern、HDR 管线或透明合成顺序不同，应重新执行上述标定，而不是直接复制 PSNR 排名。

3 评测协议

3.1 历史九场景方法矩阵

历史比较使用九类确定性场景：输入 958×442，输出 1916×884，场景时间 3.0，随机种子 17。内容包括透明实例、平面着色几何、棋盘纹理、细线、alpha cutout、高光、渐变、全屏图片和 Temple 几何。每个待测方法都与对应的 Direct 2x 帧计算 RGB PSNR。

线性插值、项目内 FSR1 EASU+RCAS、MetalFX Spatial、MSPR-Quality、MSPR-Lite 和 Direct 2x 均由旧版在同一台 iPhone 13 (iOS 18.2.1) 上生成。九个场景使用同一分辨率、时间、种子、tone map 与 RGBA8Unorm

量化契约。性能测试对每个方法和场景预热 20 帧、统计 120 帧，并以 command-buffer GPU duration 作为完整路径指标。归档 Temple 与全屏图片场景使用的资产来自 Apple “[Implementing order-independent transparency with image blocks](#)” 文档附带的可下载 Demo [1]，且仅作为测试输入。由于本项目不授予 Apple 示例资产的再分发权利，这两个场景及其资产均不随仓库发布；如需原始内容，可从所链接 Apple 官方页面自行下载并遵守 Apple 的适用条款。当前发布包只能重新生成七个项目原创程序化场景，不能重新生成这两行历史数据。

3.2 更广的完整渲染场景集

该数据集由项目自行构造并冻结，不是公开或行业通用 benchmark。脚本化实时渲染场景以 5760×3240 native-6 生成 100 帧，组成 10 个每段 10 帧的序列；随后生成 960×540 LR sample proxy 和 1920×1080 MSAA4 target。内容覆盖高对比细线与斜边、室内多层遮挡、车辆曲面高光、动态角色轮廓、半透明特效和世界空间 UI，并组合 orbit、转向、平移、冲刺、微抖动及 FOV 变化。GI、TAA、FXAA 和 post-AA 全部关闭，以隔离空间重建误差。该场景集与训练及参数标定计划的 sequence、capture session 和 content hash 重叠均为 0。

表 1: 100 帧完整渲染场景集的内容分布。

类别	帧数	相机运动	压力因素
Alias geometry	20	orbit、yaw turn	细杆、轮廓、高对比斜线
Indoor	20	sprint、FOV zoom	遮挡、硬边界、多深度层
Vehicle	20	strafe、micro-shake	高光曲线、轮胎和车身轮廓
Character	20	FOV zoom	动态轮廓、细小附件
Translucent VFX	10	sprint	alpha blending、reactive detail
World-space UI	10	yaw turn	类文字边缘、平面叠加

四个 sample 使用固定 D3D 风格位置，从 native-6 捕获中提取：以 LR 像素六分之一为单位， $(x,y) = (2,0), (5,2), (0,3), (3,5)$ 。tone-map 后的输入和 target 都量化为 RGBA8Unorm 等价精度。这个协议可复现并保留子像素相位，但不是 Metal 硬件 sample dump。

3.3 指标与输入契约

MSPR-Quality 读取四个 sample；Bicubic 和项目内逐行等价的 FSR1 EASU+RCAS 读取四 sample 的算术 resolve，FSR1 sharpness 为 0.8。Native 2× MSAA4 是参考图，不作为待评分算法。指标包含 RGB PSNR、SSIM [6]、L1、Scharr-gradient L1、edge-weighted L1、二值 edge F1 和未重投影 temporal-delta error。最后一项没有补偿相机运动，只是诊断指标。

FSR1 数字描述的是项目内 Metal 移植逻辑在本报告 positive-Reinhard/RGBA8 契约下的结果，不代表所有 AMD SDK 集成的画质排名。FSR1 文档建议输入已经过抗锯齿且位于感知颜色空间；正式游戏可能使用不同颜色管线或 RCAS 设置。

4 画质结果

4.1 历史九场景空间方案比较

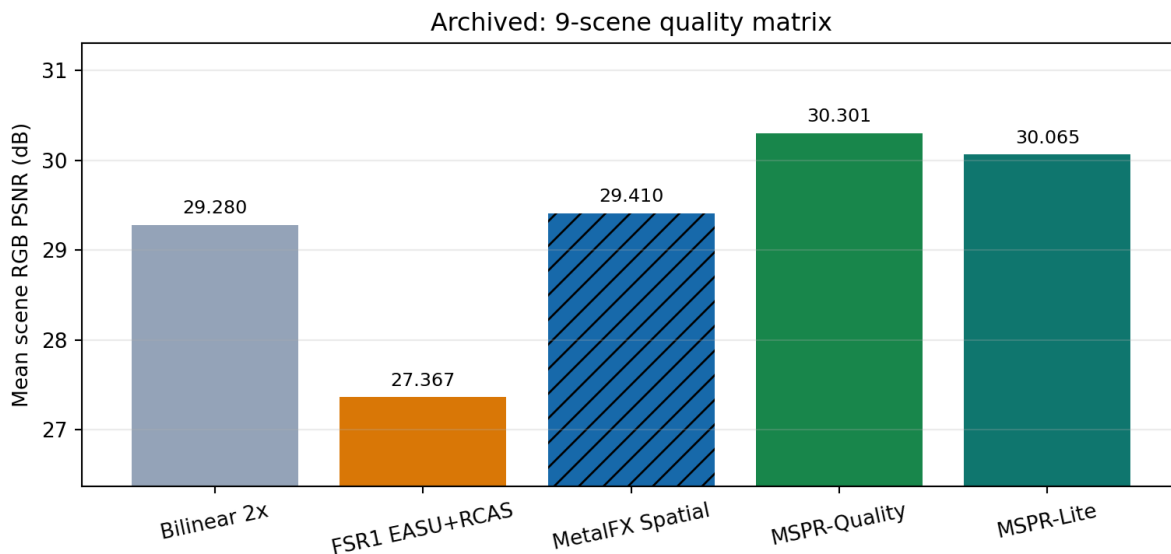


图 2: 同一台 iPhone 13 上历史九场景的平均 PSNR; 两个 Apple 依赖场景行仅作为不可复现历史证据保留。

表 2: 历史九场景方法矩阵汇总。PSNR 越高越好。

方法	场景平均 PSNR	证据来源
线性插值 2×	29.280 dB	iPhone 13 真机
FSR1 EASU+RCAS	27.367 dB	iPhone 13 真机
MetalFX Spatial	29.410 dB	iPhone 13 真机
MSPR-Quality	30.301 dB	iPhone 13 真机
MSPR-Lite	30.065 dB	iPhone 13 真机
Direct 2×	参考图	iPhone 13 真机

在历史矩阵中，MSPR-Quality 的算术场景均值比 MSPR-Lite 高 0.237 dB，比 MetalFX 高 0.891 dB，比线性插值高 1.021 dB，比 FSR1 高 2.934 dB；它在 transparency、specular 和 Temple 中取得最高 PSNR。MSPR-Lite 在 flat shading 和 gradient 中领先，并以接近 Quality 的总体画质提供更低成本。Temple 结论仅为历史结果，不能由当前发布包重跑。

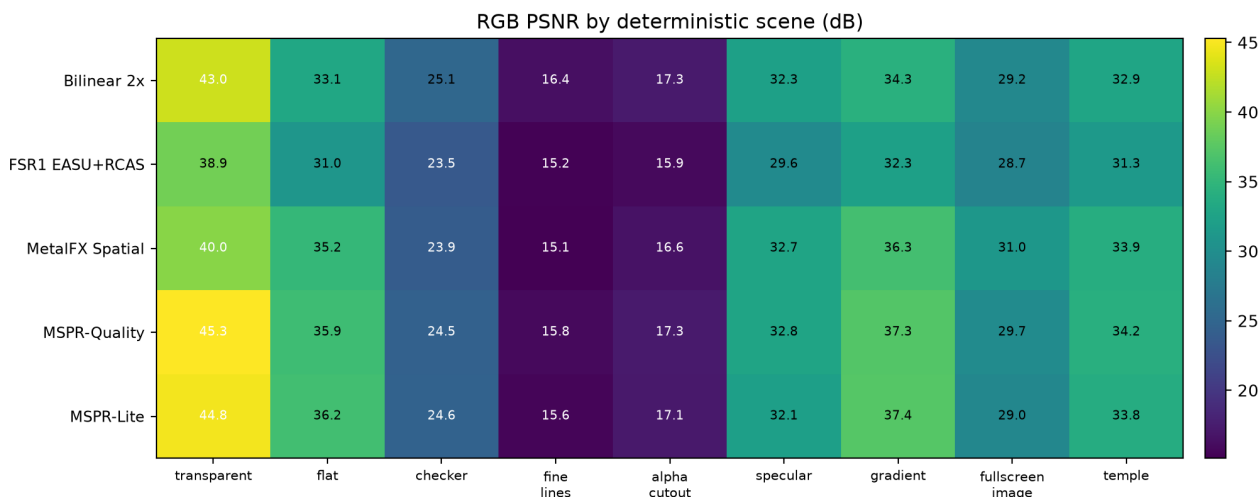


图 3: 九类内容的逐场景 RGB PSNR; 该矩阵展示了单个均值无法反映的优势与薄弱项。

4.2 当前发布版典型 case

三组发布版 shader 案例覆盖透明实例、alpha cutout 和 specular 内容, 时间为 3.0、种子为 17。每张图都包含同一台 iPhone 13 生成的 LR resolve、线性插值、FSR1、MetalFX Spatial、两个 MSPR 配置、Direct 2x、固定局部放大、各方法 PSNR, 以及放大五倍的 MSPR-Lite 误差图。

发布包可以从源码重新生成七场景协议的数值和图片。先用 Xcode 将 MSPR-Quality-iOS 安装到物理 iPhone, 然后在仓库根目录执行:

```
uv sync --frozen
uv run python Demo/Tools/reproduce_release_quality.py --device <iPhone-UDID>
```

脚本依次运行七个已分发场景, 将质量矩阵写入 `Reproduction/MSPR-Quality/quality-matrix.json`, 从 App Documents 拉取代表性 PPM/PNG, 并在 `figures/` 生成汇总图和案例图。完整 MetalFX 对比需要物理 iPhone; 该命令不会生成被移除的两个历史场景行。

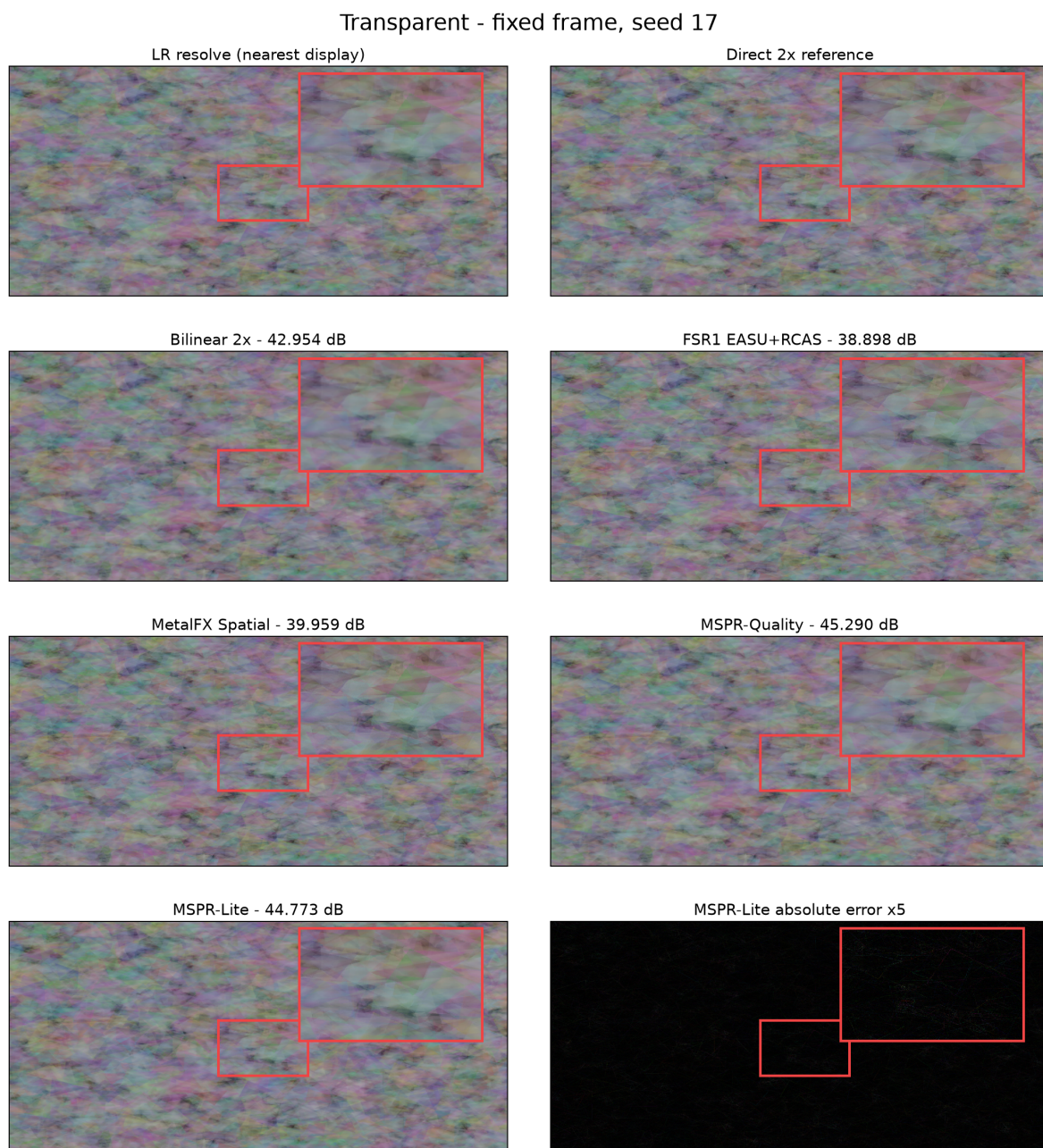


图 4: 透明压力场景。MSPR 保留 resolve 前相位证据, 并在该场景取得最高 PSNR。

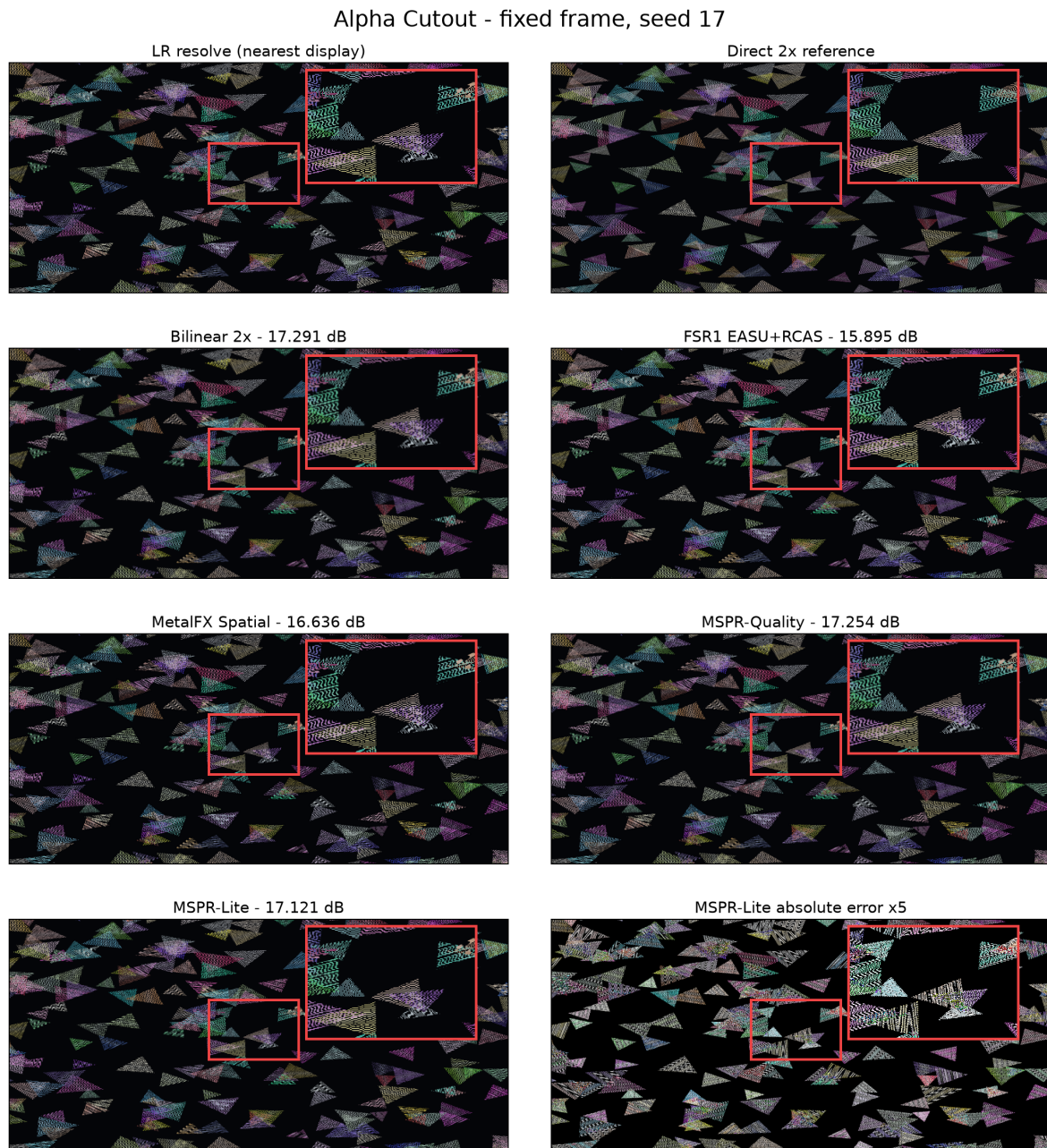


图 5: Alpha-cutout 场景。线性插值只比 MSPR-Quality 高 0.037 dB，局部放大图显示了两者不同的边缘响应。

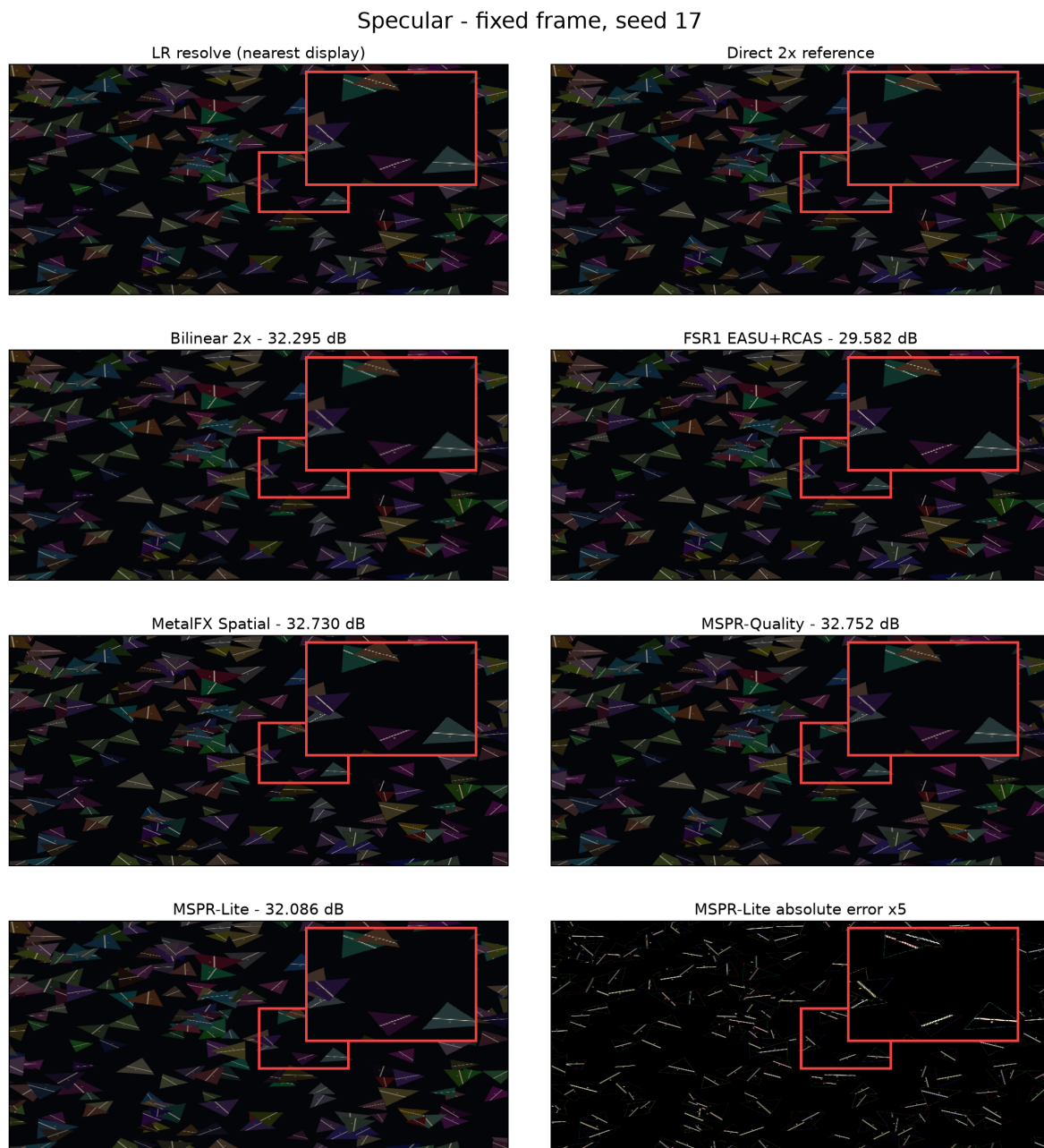


图 6: Specular 场景。MSPR-Quality 的 PSNR 最高，比 MetalFX Spatial 高 0.022 dB。

4.3 更广的 100 帧完整渲染场景集

该离线场景集不能运行 MetalFX，也没有 MSPR-Lite 输出，因此它只作为 MSPR-Quality、Bicubic 和 FSR1 的补充证据，不参与主要方法排名。

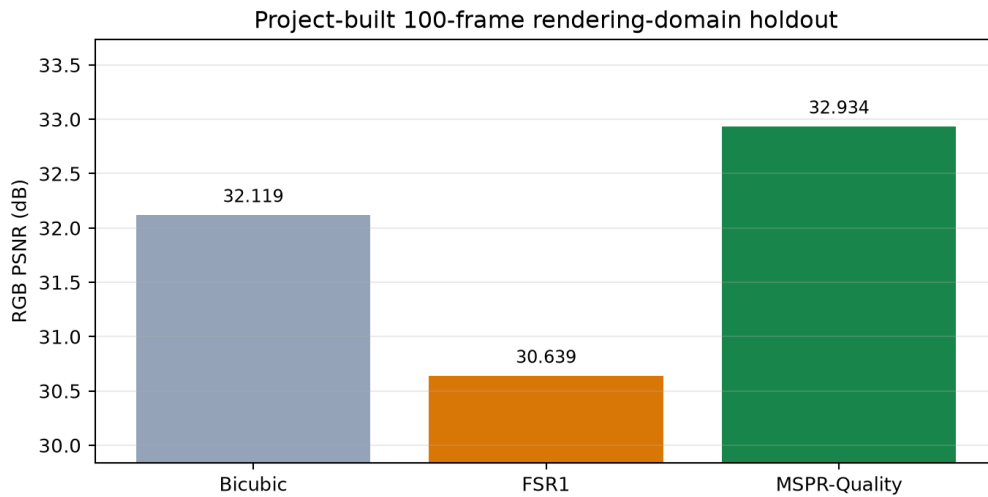


图 7: 更广的 100 帧完整渲染场景集, PSNR 越高越好。

表 3: 更广场景集的平均结果。箭头表示指标更优的方向。

方法	PSNR↑	SSIM↑	L1↓	Grad. L1↓	Edge F1↑
Bicubic resolve	32.119	0.97949	0.00444	0.00700	0.8258
FSR1 EASU+RCAS	30.639	0.97474	0.00563	0.00763	0.7858
MSPR-Quality	32.934	0.98504	0.00351	0.00521	0.8770

MSPR-Quality 相对 Bicubic 的 0.815 dB 增益同时伴随 L1 和梯度误差降低。相对 FSR1 的 2.295 dB 优势必须与输入契约一起阅读: FSR1 无法使用已经在 resolve 中丢失的相位证据。

4.4 完整实时渲染场景结果

这些结果来自完整渲染序列, 而不是孤立的程序化图案。内容覆盖高锯齿压力几何、车辆曲面高光与细几何, 以及带相机运动的半透明 VFX, 用于补充上面的固定 release-shader 场景矩阵。

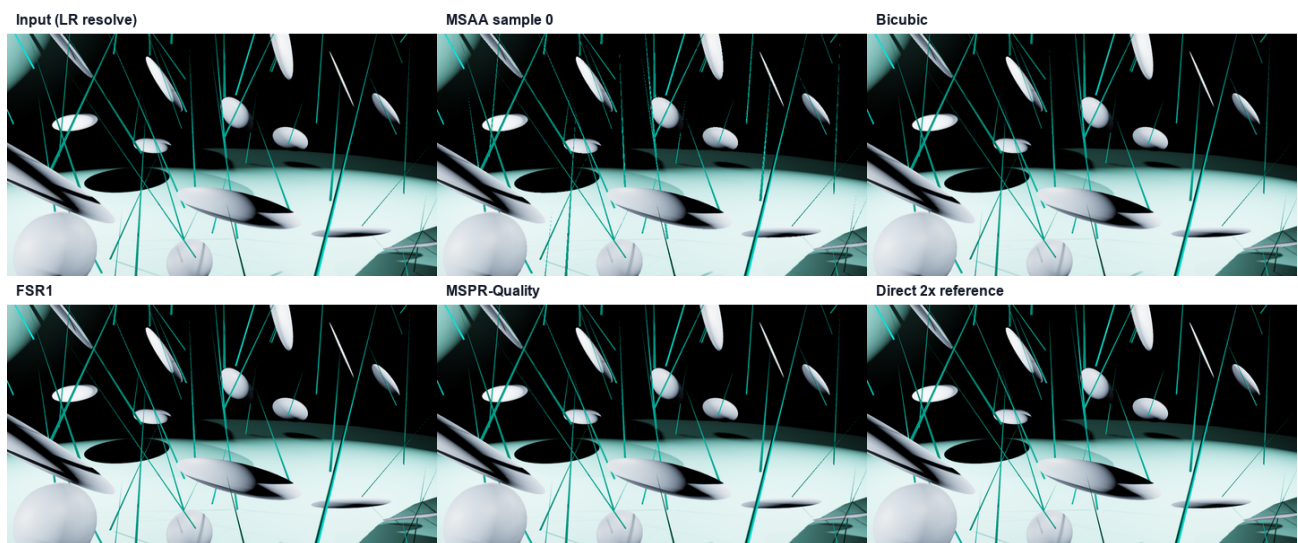


图 8: Alias-stress case: LR resolve、一个 resolve 前 MSAA sample、Bicubic、FSR1、MSPR-Quality 和 native 2× reference。

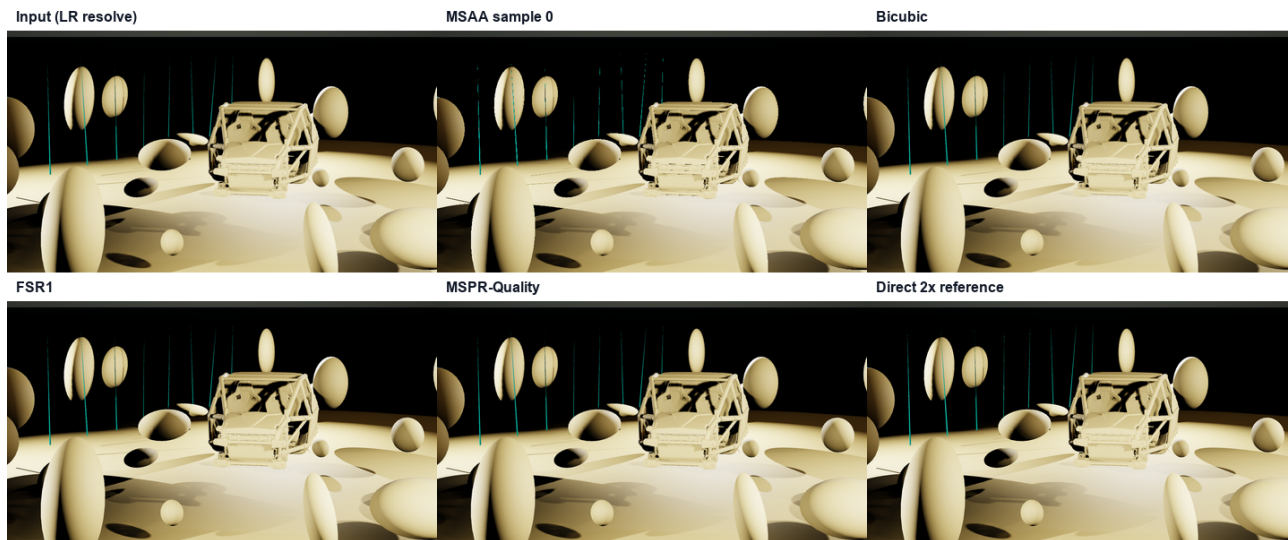


图 9: Vehicle case: 曲面高光边界与细几何。

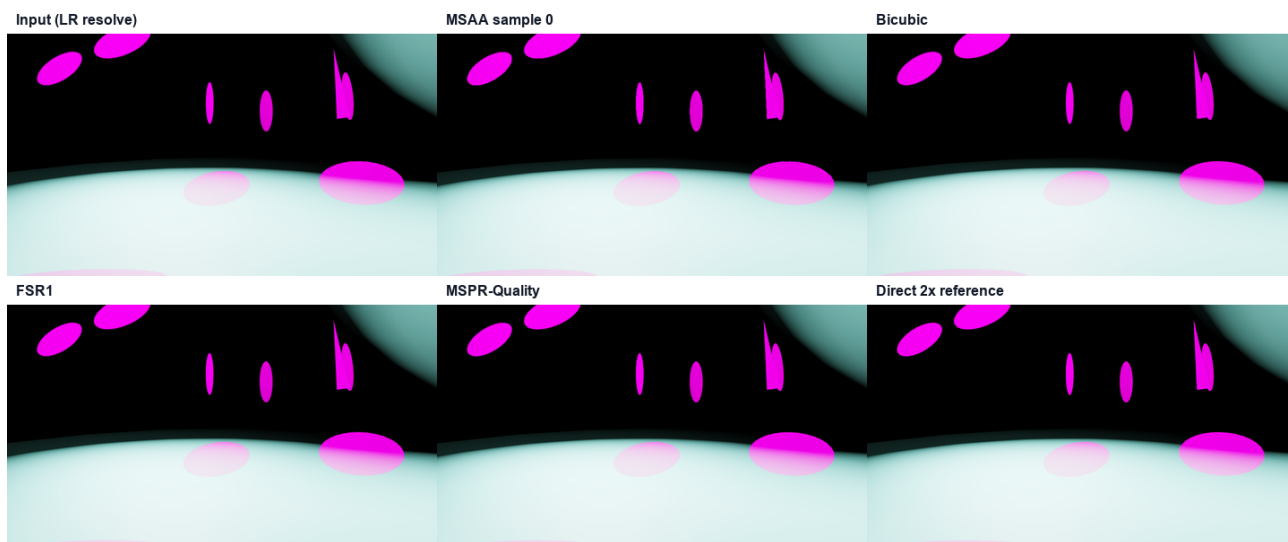


图 10: 透明 VFX case。即使总体指标改善，alpha-blended 内容仍是部署风险。

5 性能与带宽证据

5.1 设备与测量方法

历史测试设备为 iPhone 13 (Apple A15 GPU), 系统为 iOS 18.2.1; 输入分辨率 958×442 , 输出分辨率 1916×884 。9 个场景均使用 scene time 3.0 和 seed 17; 每种方法在每个场景预热 20 帧, 随后统计 120 帧。GPU 时间是完整 command buffer 的 duration, 不是 CPU 编码时间, 也不是独立 shader kernel 时间。方法均值是在与 PSNR 相同的九个场景上等权算术平均。

5.2 同一矩阵上的完整路径对比

表 4: iPhone 13 历史九场景矩阵上的画质与完整路径 GPU 时间（GPU 时间越低越好）。

方法	RGB PSNR 均值	GPU 均值 (ms)	相对 Bilinear 增量 (ms)
MSPR-Quality	30.301	3.656	1.067
MSPR-Lite	30.065	3.218	0.629
MetalFX Spatial	29.410	7.606	5.017
Bilinear 2×	29.280	2.589	0.000 (基准)
FSR1 EASU+RCAS	27.367	8.575	5.986

最后一列采用单一且透明的参考：

$$\Delta_{\text{full}}(m) = \bar{T}_{\text{GPU}}(m) - \bar{T}_{\text{GPU}}(\text{Bilinear}), \quad (7)$$

其中每个均值都是九场景等权平均。这是同一负载契约下完整渲染路径的差值；不再减去单独的 Formula-only baseline，也不能表述为独立重建 kernel 时间。按这一口径，MSPR-Quality 以 1.067 ms 的完整路径增量提升 1.021 dB；MSPR-Lite 以 0.629 ms 的增量提升 0.785 dB。同一矩阵中，MetalFX Spatial 与 FSR1 的增量分别为 5.017 ms 和 5.986 ms，明显更高。

MSPR 将 MSAA sample 和 resolved neighborhood 保留在 tile memory 中，只持久化最终 2× 图。FSR1 通常需要持久化 LR resolve，并在 RCAS 之前写入和读取全分辨率 EASU 中间图。SGSR1 将上采样和锐化压到一个 pass，但仍从 resolved color 开始。MetalFX 暴露输入/输出纹理，内部流量不透明。逻辑 pass 数并不等于 DRAM 字节数，Apple 压缩、cache、store action 和 tile spill 才决定真实带宽。

这些完整路径均值同时包含场景渲染与重建，因此适合在相同负载下比较部署方案，但不能据此推导独立 kernel 成本。两个依赖 Apple Demo 的场景贡献属于历史证据；新部署应使用可再分发、能代表自身 renderer 的内容重跑完整矩阵。

6 与主流移动上采样器的关系

表 5: 不同方法的输入契约与本报告提供的证据。

方法	输入	执行方式	主要优势	本报告证据
MSPR-Quality	四个 MSAA color + coverage/neighborhood	tile-local spatial	保留 resolve 前相位	iPhone 13 画质与完整路径时间
MSPR-Lite	四个 MSAA color + 左右 resolved tap	无 barrier 的 tile-local spatial	较小完整路径增量	iPhone 13 画质与完整路径时间
FSR1	resolved color	EASU + RCAS	可移植的开源空间基线	iPhone 13 画质与时间
SGSR1	resolved color	单个移动 shader	面向 Adreno 带宽	仅设计对比
MetalFX Spatial	color texture	Apple runtime	平台集成	iPhone 13 画质与时间
SGSR2	color、depth、motion、history	temporal	动态画质与稳定性	非等价契约
Arm ASR	FSR2 类 temporal 输入	temporal mobile	开源跨 GPU temporal	非等价契约
Direct 2×	native scene render	全分辨率 raster	参考画质	GT；iPhone 时间

SGSR2 与 Arm ASR 能利用跨帧信息，目标包含静态 MSPR 无法解决的锯齿闪烁 [4, 7, 8]；代价是 motion、depth、jitter、history 和 disocclusion 处理。因此公平比较应把 MSPR 与相同无历史工作点的空间 post-resolve 方法对比，temporal 方法作为另一个画质层级单独报告。

A 照片 proxy: 跨域可扩展性观察

Set5、Set14 与 Urban100 都是照片，不是渲染场景。因此它们不参与 MSPR-Quality 参数选择，也不进入主上线结论。它们只用于观察：在合成四 sample proxy 下，相位重建能否适应重复结构与自然图像统计 [9, 10, 11]。

表 6: 跨域 RGB PSNR (dB)。不是标准 single-LR SISR 排行榜。

方法	Set5	Set14	Urban100
Bicubic resolve	31.207	28.144	25.123
FSR1 resolve	28.181	25.946	23.405
MSPR 照片配置	37.567	34.948	32.247

照片 proxy 的大幅增益说明算法具有可扩展性，而不是说明照片配置可以用于渲染。该配置使用更强负 lobe，在渲染 internal validation 上低于 MSPR-Quality (33.512 对 36.135 dB)。因此参数应按内容域管理。

B 可复现性与局限

发布包保留参数记录、逐帧 CSV、汇总 JSON、当前发布版场景原始日志，以及完整渲染场景 PNG。当前场景矩阵用于选择发布版默认配置，因此它作为发布验证证据报告，而不是未参与选择的独立 benchmark。评估器采用确定性的 positive-Reinhard tone map 与 RGBA8Unorm 等价值量化。

仍存在的有效性威胁包括：更广场景集使用空间 sample proxy、GI-off 内容、缺少正式 post-processing、项目特定的 FSR1 颜色契约、没有 motion-compensated temporal metric、完整路径时间不能替代独立 kernel 时间、九场景均值包含两个依赖 Apple Demo 的历史行，以及没有 SGSr 画质输出。报告明确陈述这些限制，不推断等价性。

参考文献

- [1] Apple Developer Documentation. Implementing order-independent transparency with image blocks. <https://developer.apple.com/documentation/metal/implementing-order-independent-transparency-with-image-blocks>. Accessed 2026-08-17.
- [2] AMD GPUOpen. Fidelityfx super resolution 1, 2021.
- [3] AMD GPUOpen. Fidelityfx-fsr: Fsr1 reference source, 2021.
- [4] Snapdragon Game Studios. Snapdragon game super resolution, 2024.
- [5] Apple Developer Documentation. Mtlfxspatialscaler.
- [6] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [7] Arm. Introducing arm accuracy super resolution, 2024.
- [8] Arm. Arm accuracy super resolution, 2025.
- [9] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, 2012.
- [10] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730. 2010.
- [11] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015.