Description

This is a supplementary appendix for the paper:
"Adaptive multilinear tensor product wavelets"
by Kenneth Weiss and Peter Lindstrom.

It contains experimental results from our testbed of 2D and 3D scalar fields over regular grids. For each dataset:

- Figure X.1 compares the adaptive and uniform meshes in terms of the number of wavelets, vertices and supercubes using the Peak Signal to Noise Ratio (PSNR).
- Figure X.2 compares the efficiency of our adaptive representation. Here we plot wavelets against the number of vertices per wavelet (top), where higher is better. For the middle and bottom charts, we investigate the efficiency of the different stencils listed in the paper and report results relative to the largest stencil ('Full Rectilinear Stencil,' which includes all samples in the support of each wavelet).
- Figure X.3 is the same as the top of Figure X.2, but for saturated wavelets. It shows that in our hierarchical multilinearly and radially interpolated meshes, many of the support vertices (i.e. those included only to satisfy hierarchical and stencil support dependencies) have their full support in the mesh. Thus, we can increase the fidelity of our reconstructed functions without any additional runtime costs by incorporating their associated wavelet coefficients in our reconstruction.
- Finally, Figure X.4 investigates the benefits of using the native multilinear interpolant induced by the subdivision scheme rather than a linear interpolant over a triangulated (tetrahedralized) mesh. The latter is generated from the same number of wavelets as the former, but requires some additional vertices (and supercubes) to triangulate (tetrahedralize) the mesh.

We extracted meshes using many different thresholds of the wavelet coefficients (expressed as percentages of the original mesh vertices), but, for clarity, we only add markers to the charts for meshes extracted using $1 \cdot 10^N$ and $3 \cdot 10^N$ percent of the coefficients (e.g. 0.001%, 0.003%, 0.01%, ...).

The PSNR charts have dark horizontal gridlines every 3 dB, with faint minor gridlines every 0.5 dB.

We note that our depth-based meshes are all saturated (Type II dependencies, where we include wavelet coefficients of all wavelets with full support in the mesh). Thus, the comparisons with respect to their unsaturated adaptive counterparts are meant only for reference.

Part I

2D

Chapter 1

Lena (513^2)

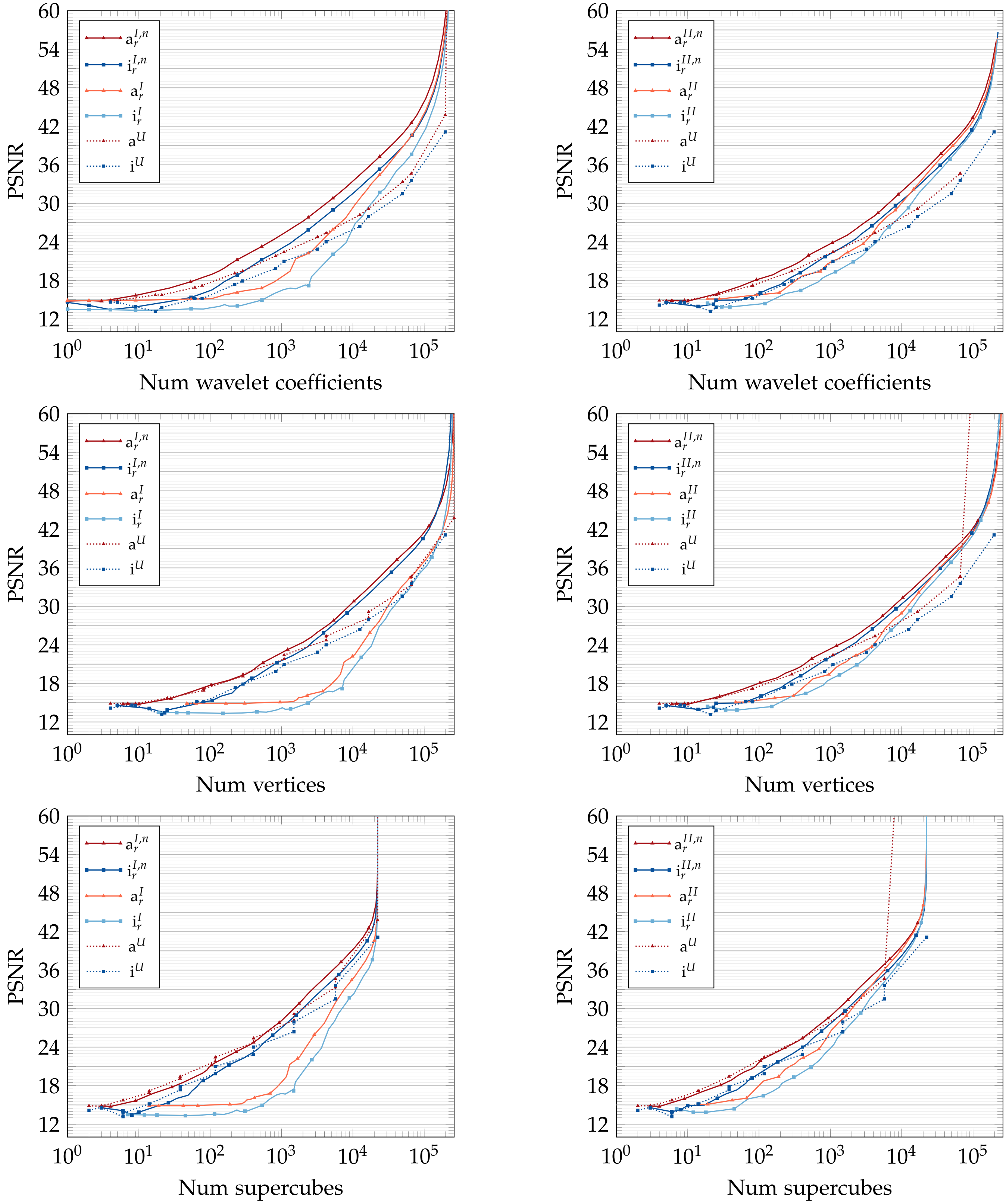

Figure 1.1: Lena (513^2) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

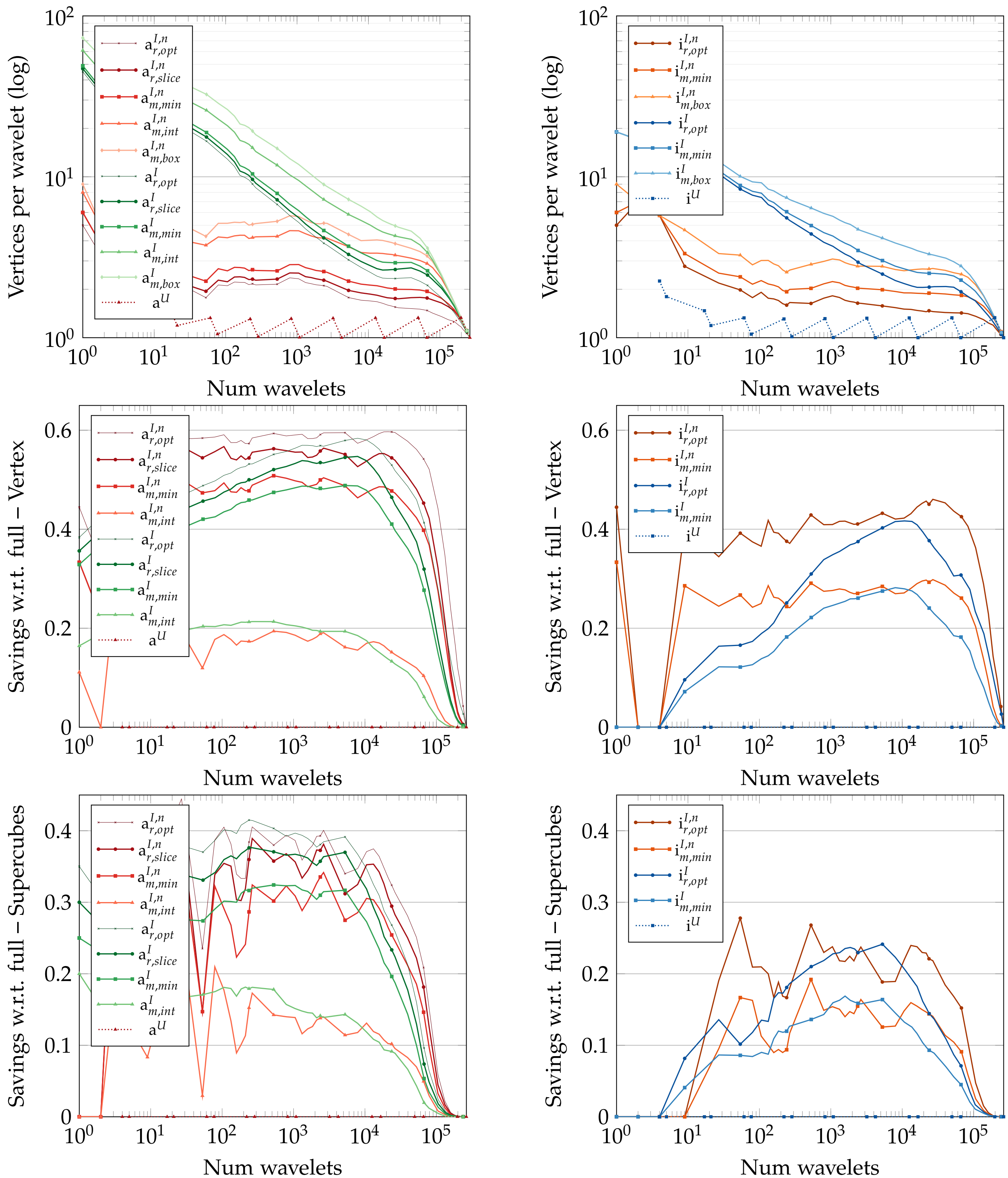

Figure 1.2: Lena (513^2) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil} / vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil} / super_{fullGrid})$.

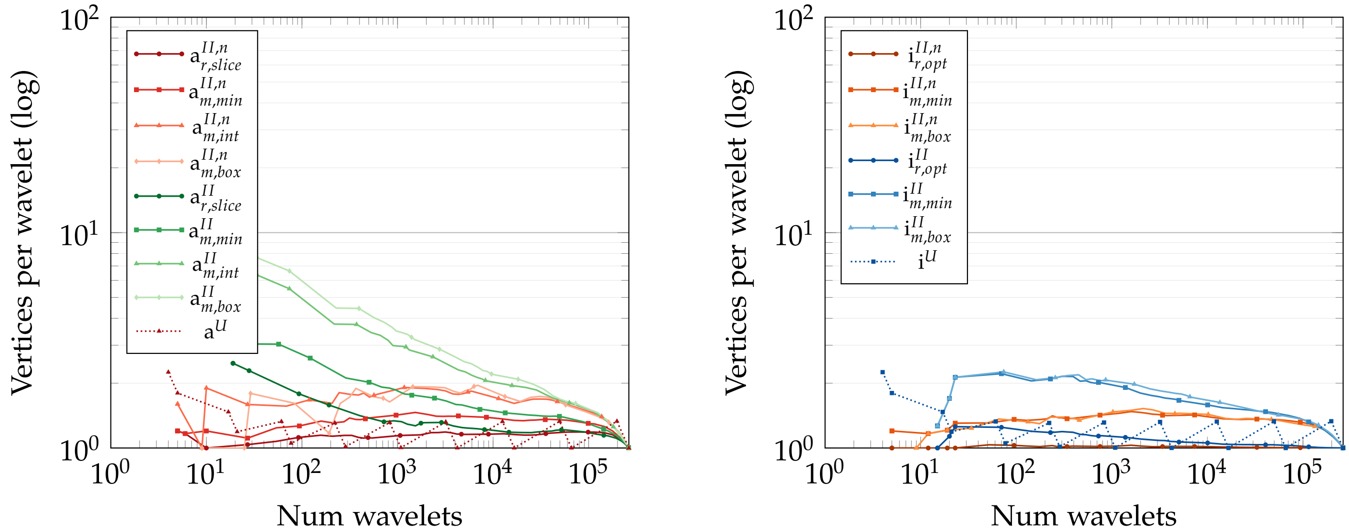

Figure 1.3: Lena (513^2) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

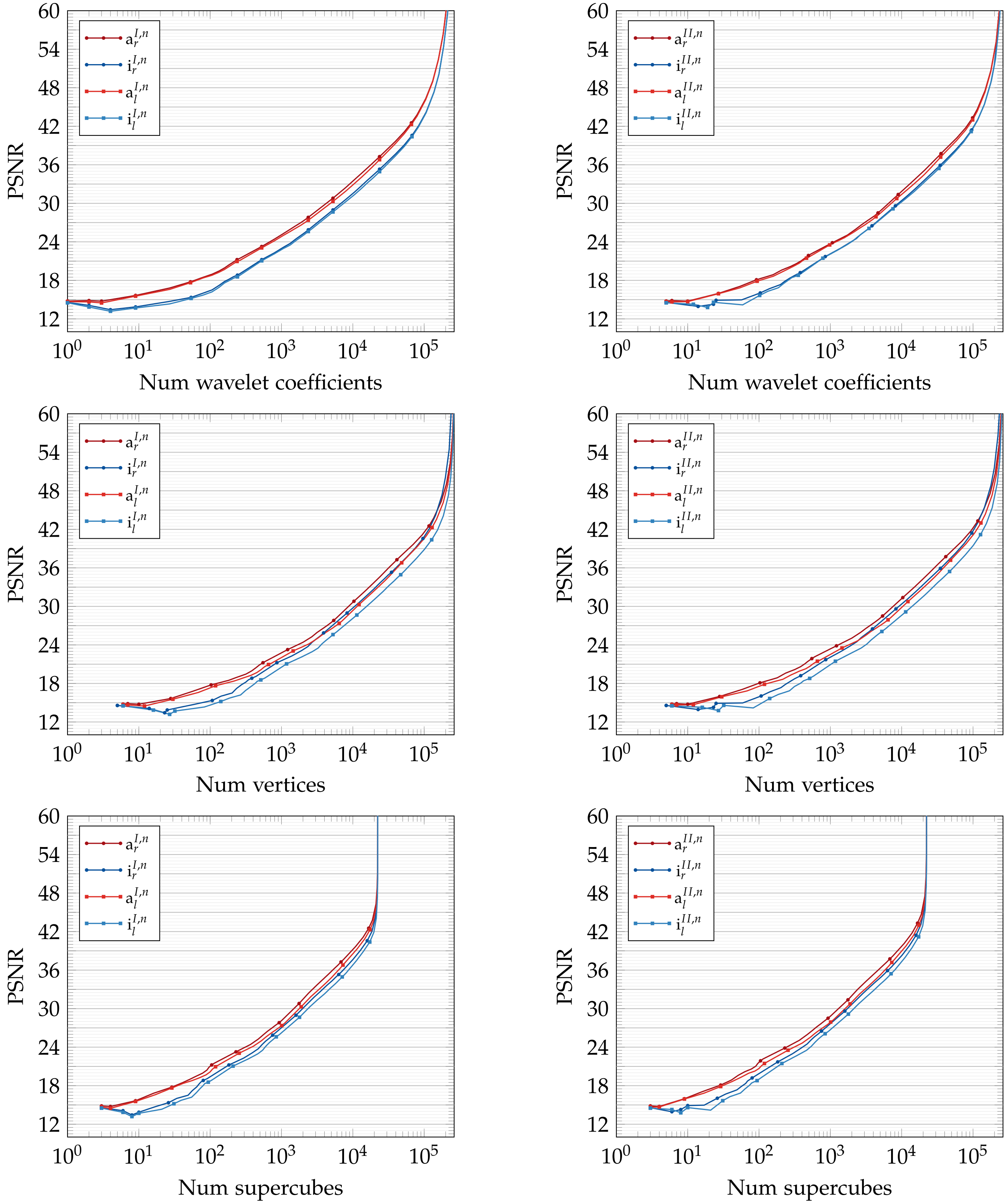

Figure 1.4: Lena (513^2) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 2

Barbara (513^2)

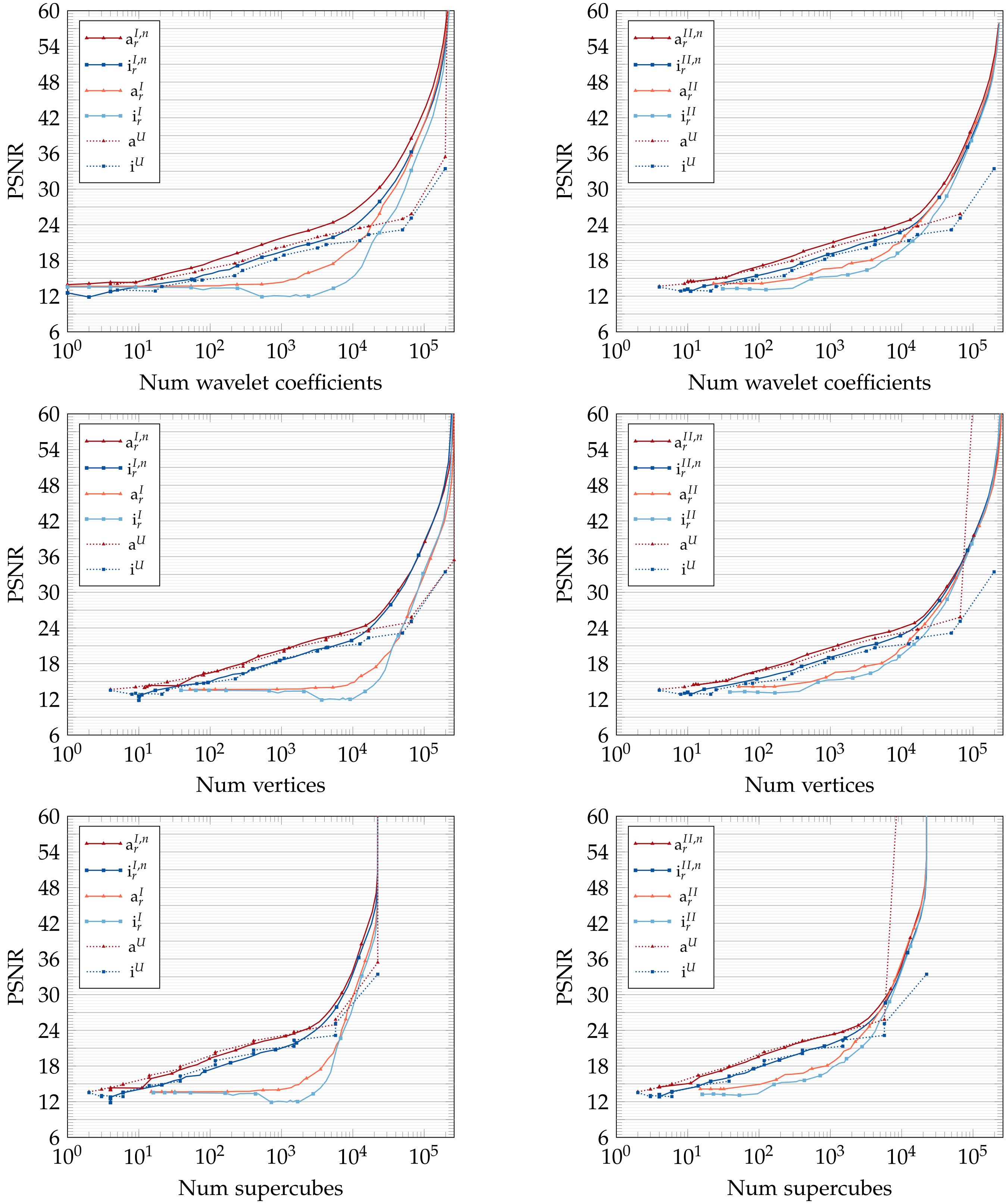

Figure 2.1: Barbara (513^2) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

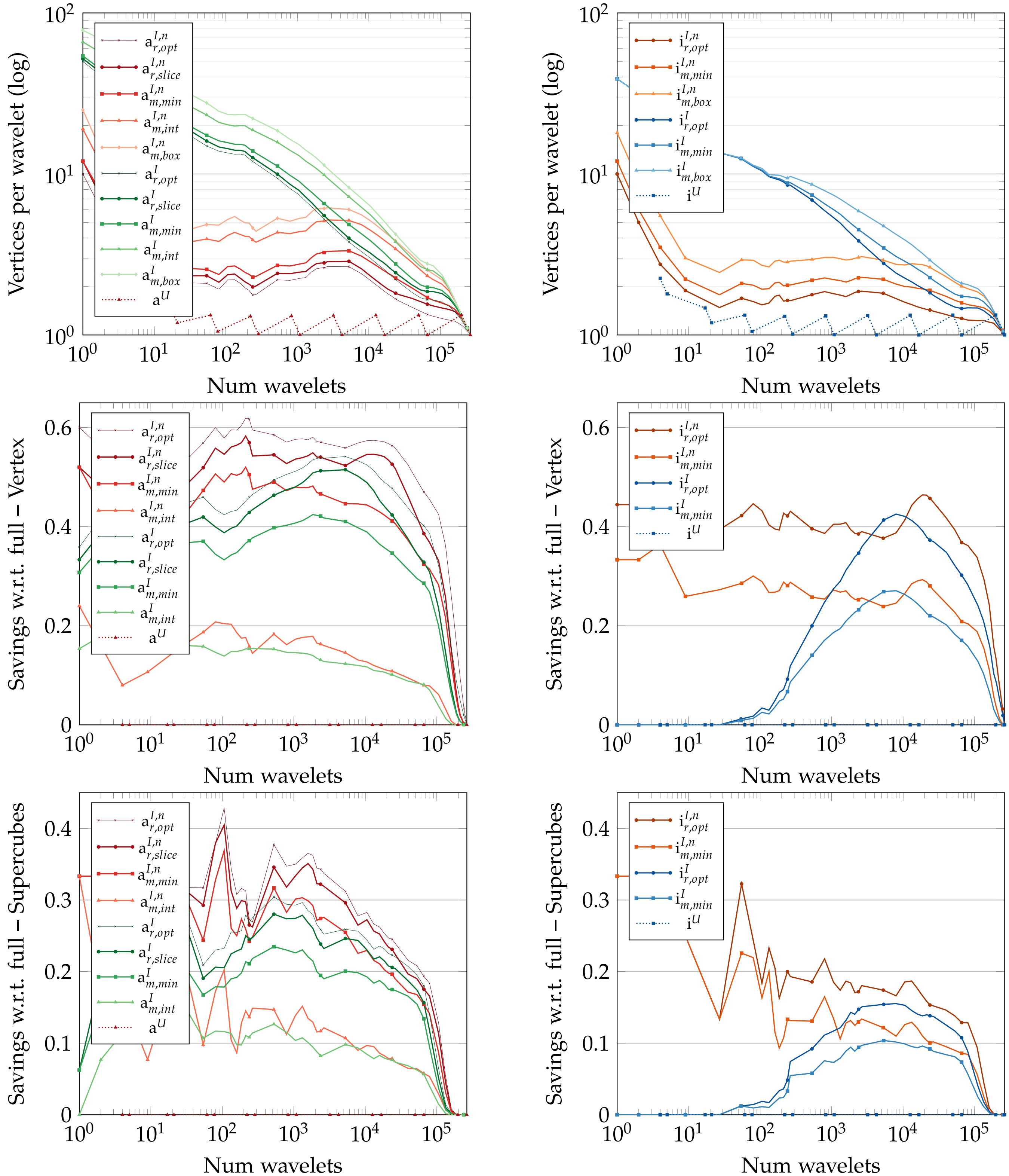

Figure 2.2: Barbara (513^2) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil}/vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil}/super_{fullGrid})$.

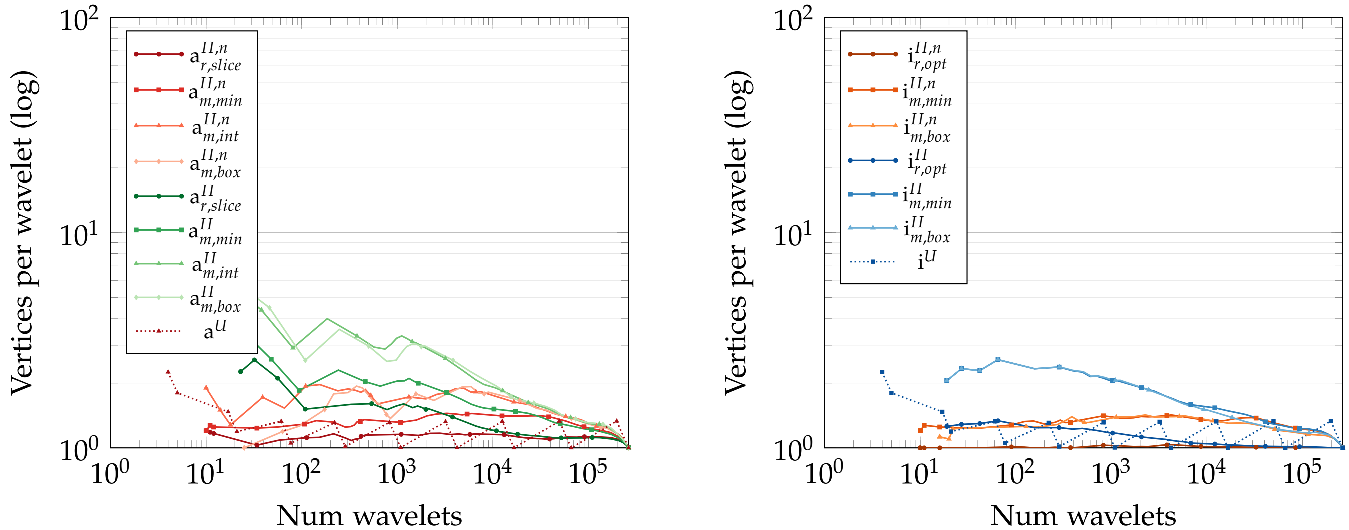

Figure 2.3: Barbara (513^2) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

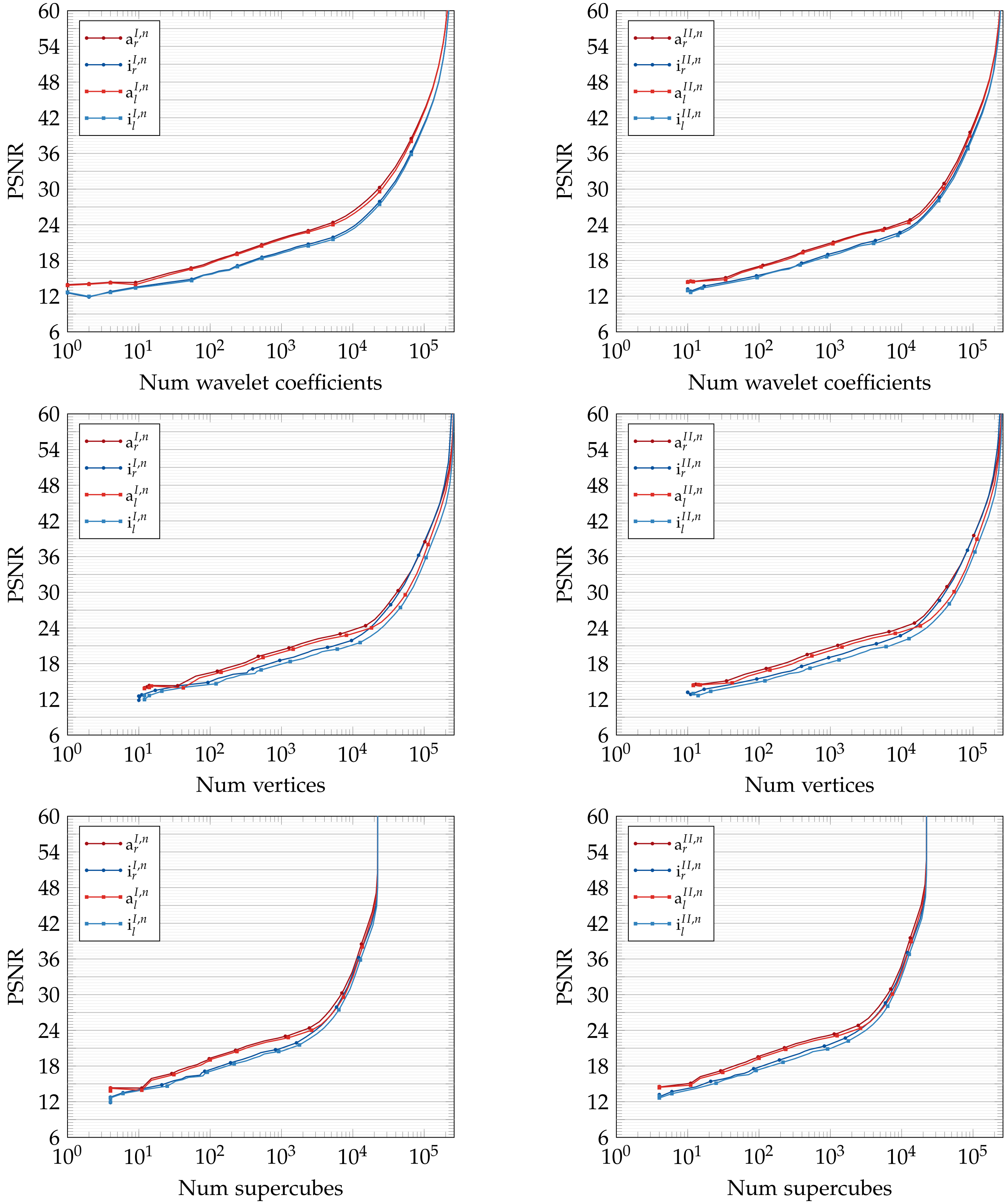

Figure 2.4: Barbara (513^2) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 3

Bike (2049^2)

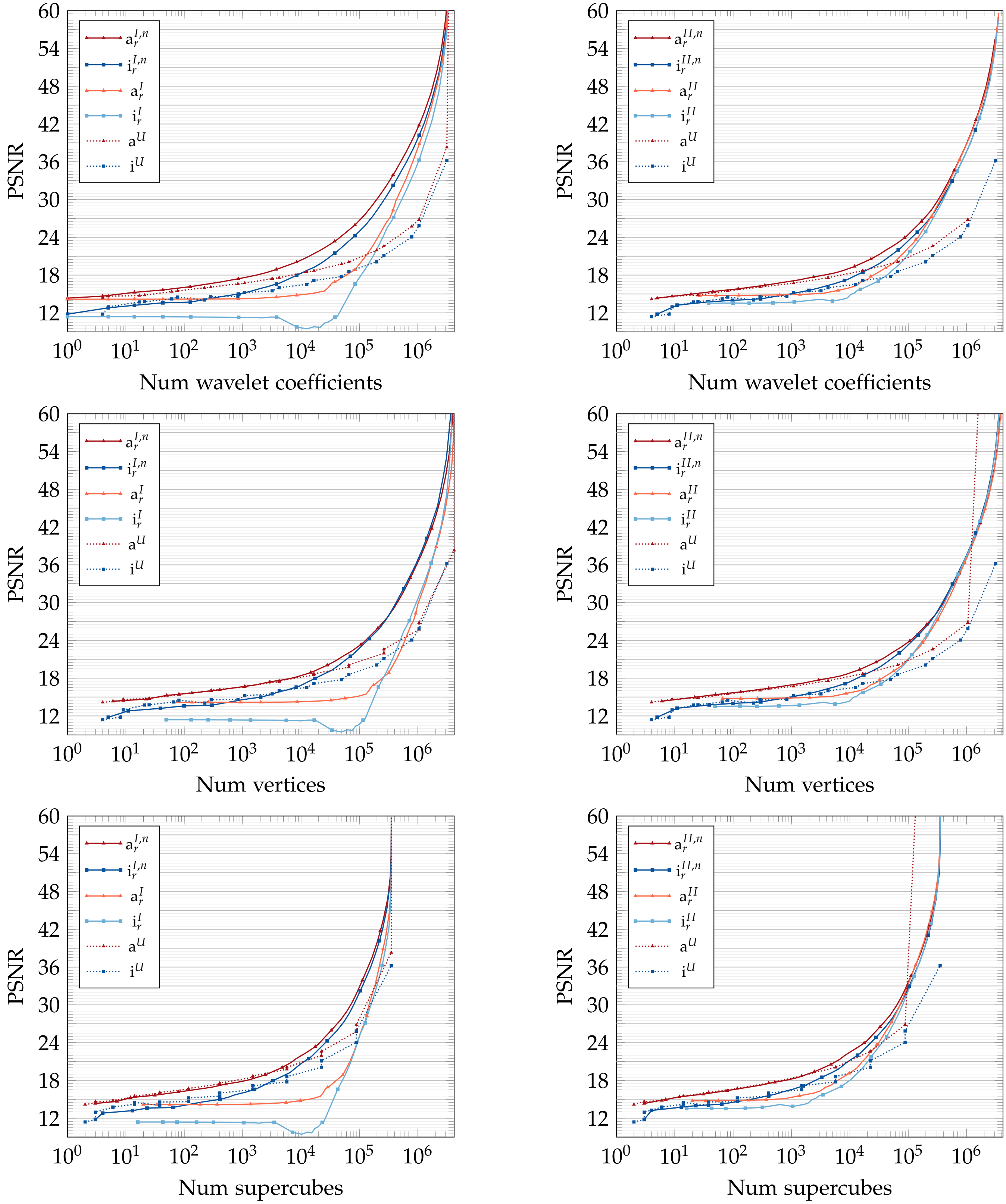

Figure 3.1: Bike (2049^2) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

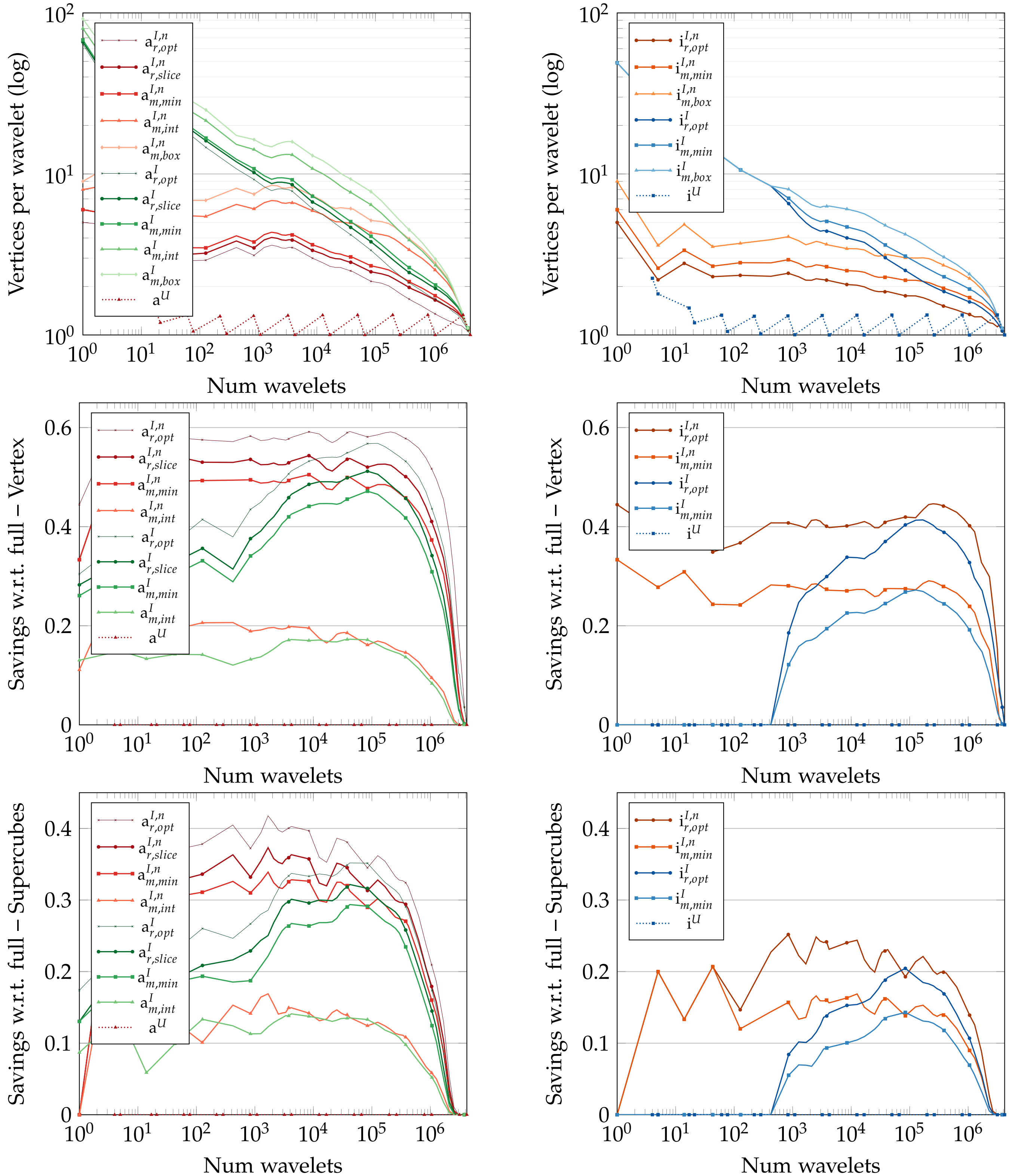

Figure 3.2: Bike (2049^2) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1-(vert_{currentStencil}/vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1-(super_{currentStencil}/super_{fullGrid})$.

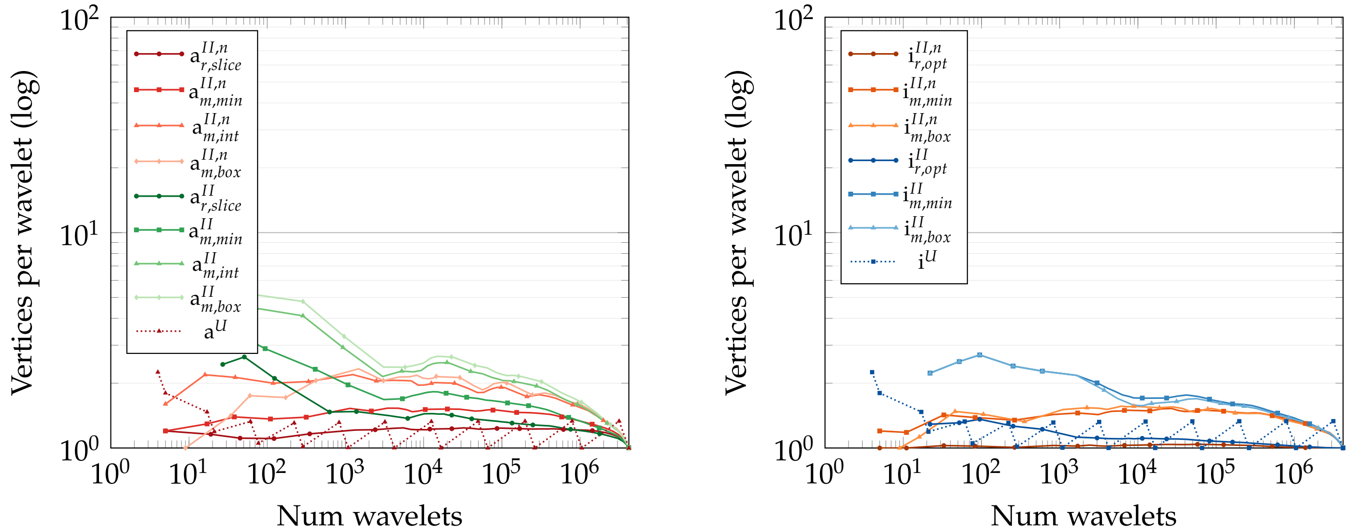

Figure 3.3: Bike (2049^2) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

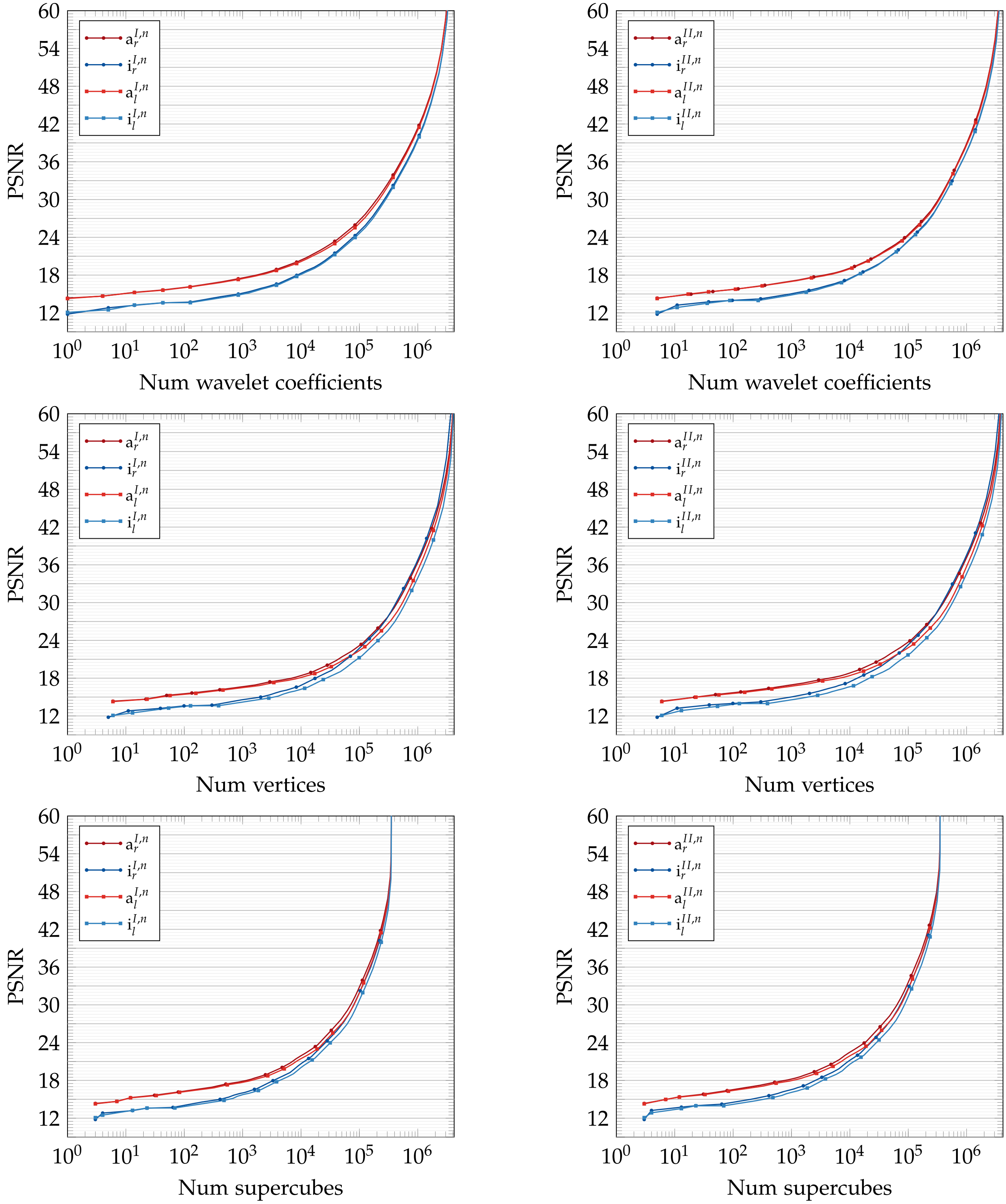

Figure 3.4: Bike (2049^2) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 4

Puget 4K (4097^2)

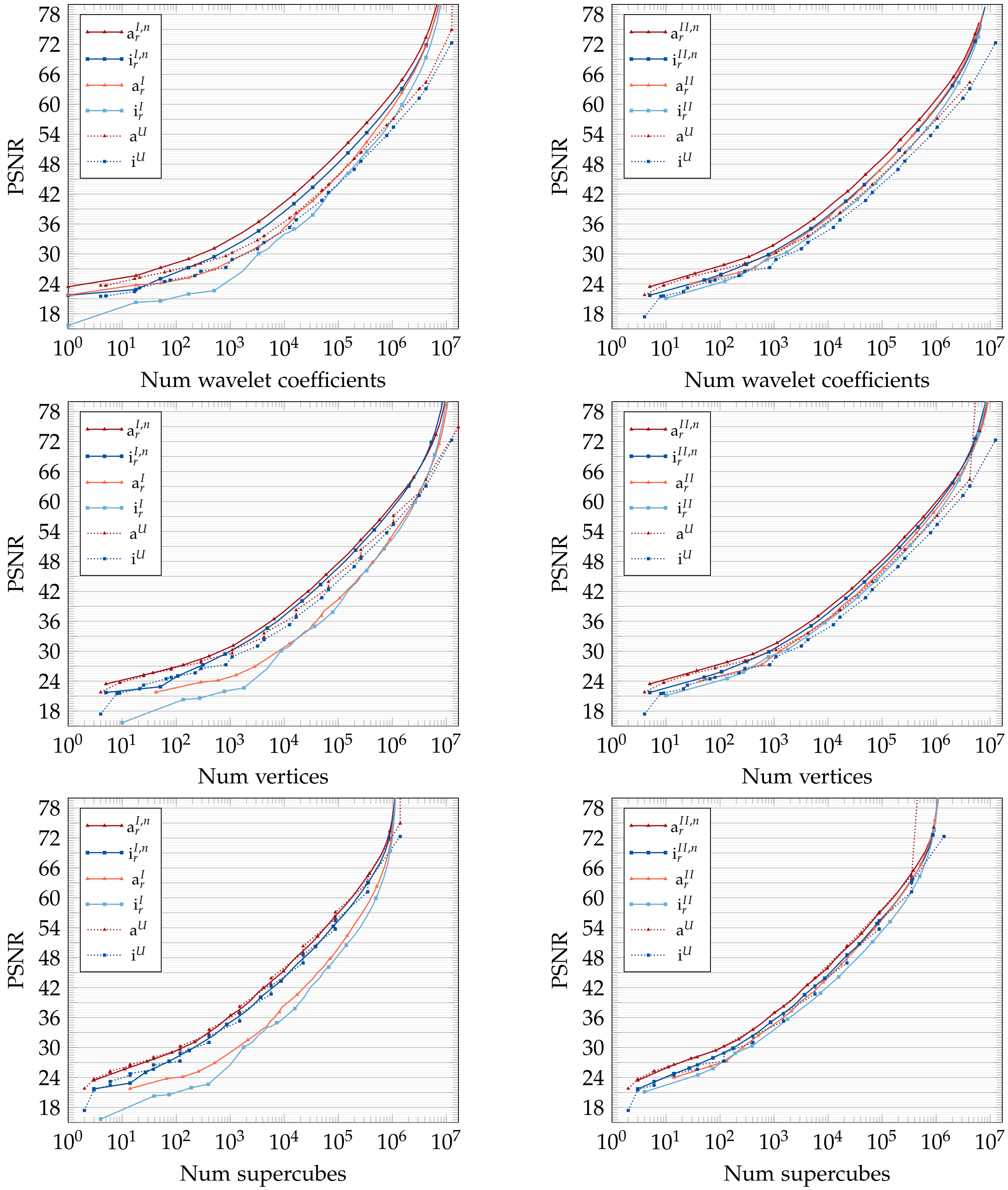

Figure 4.1: Puget 4K (4097^2) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

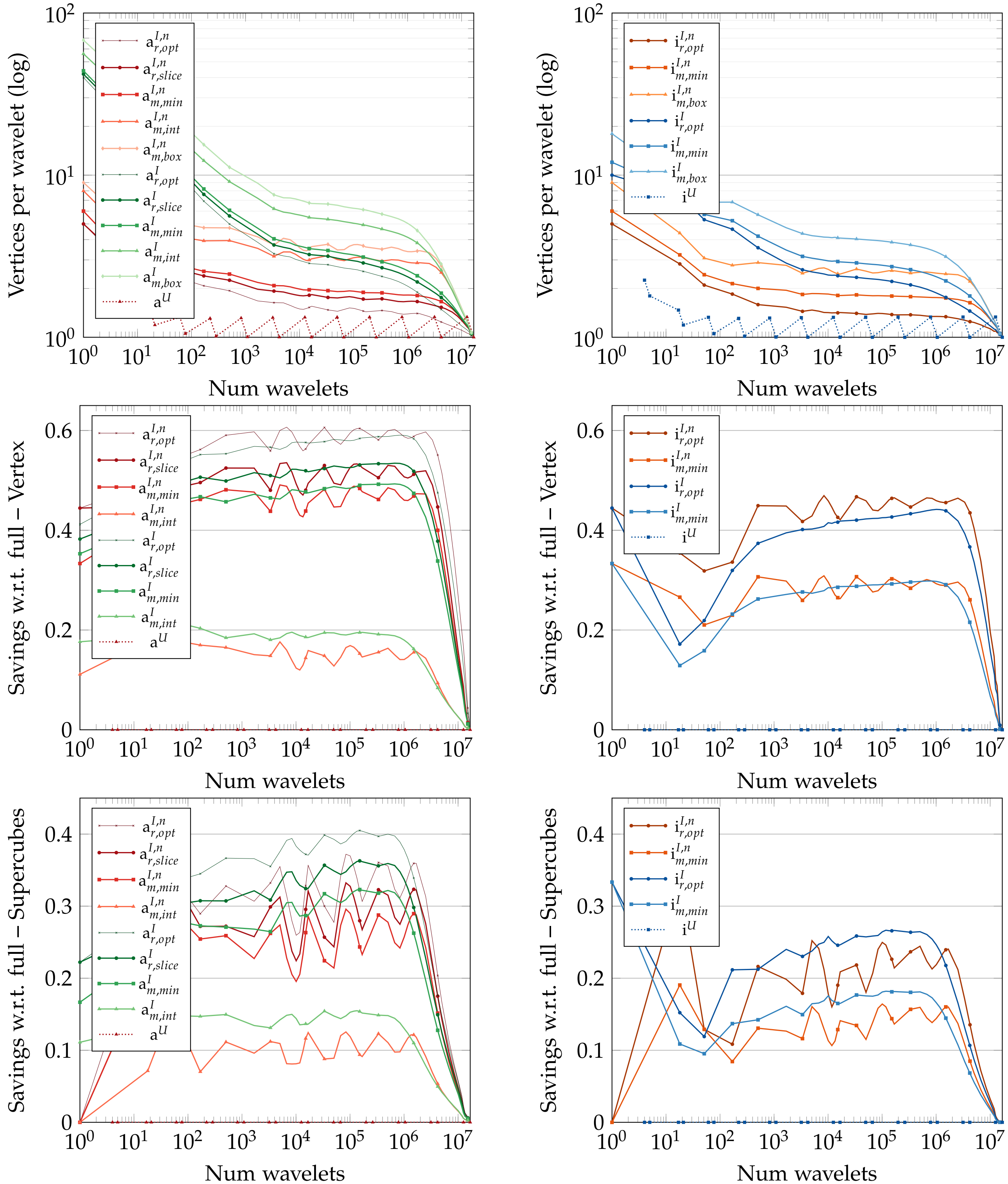

Figure 4.2: Puget 4K (4097^2) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil} / vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil} / super_{fullGrid})$.

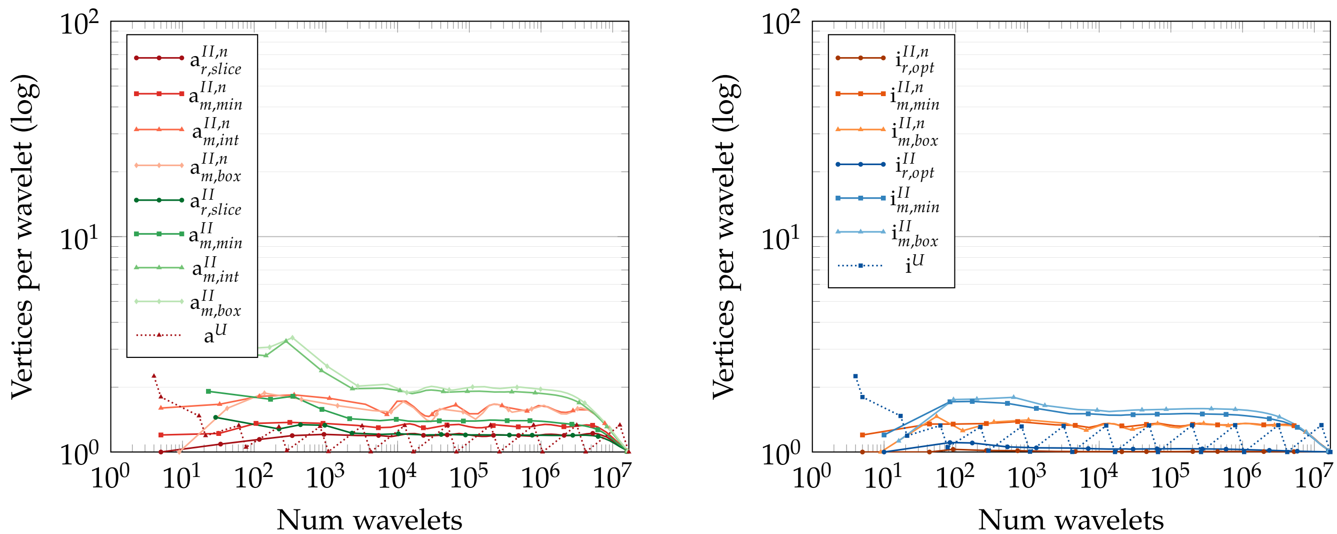

Figure 4.3: Puget 4K (4097^2) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

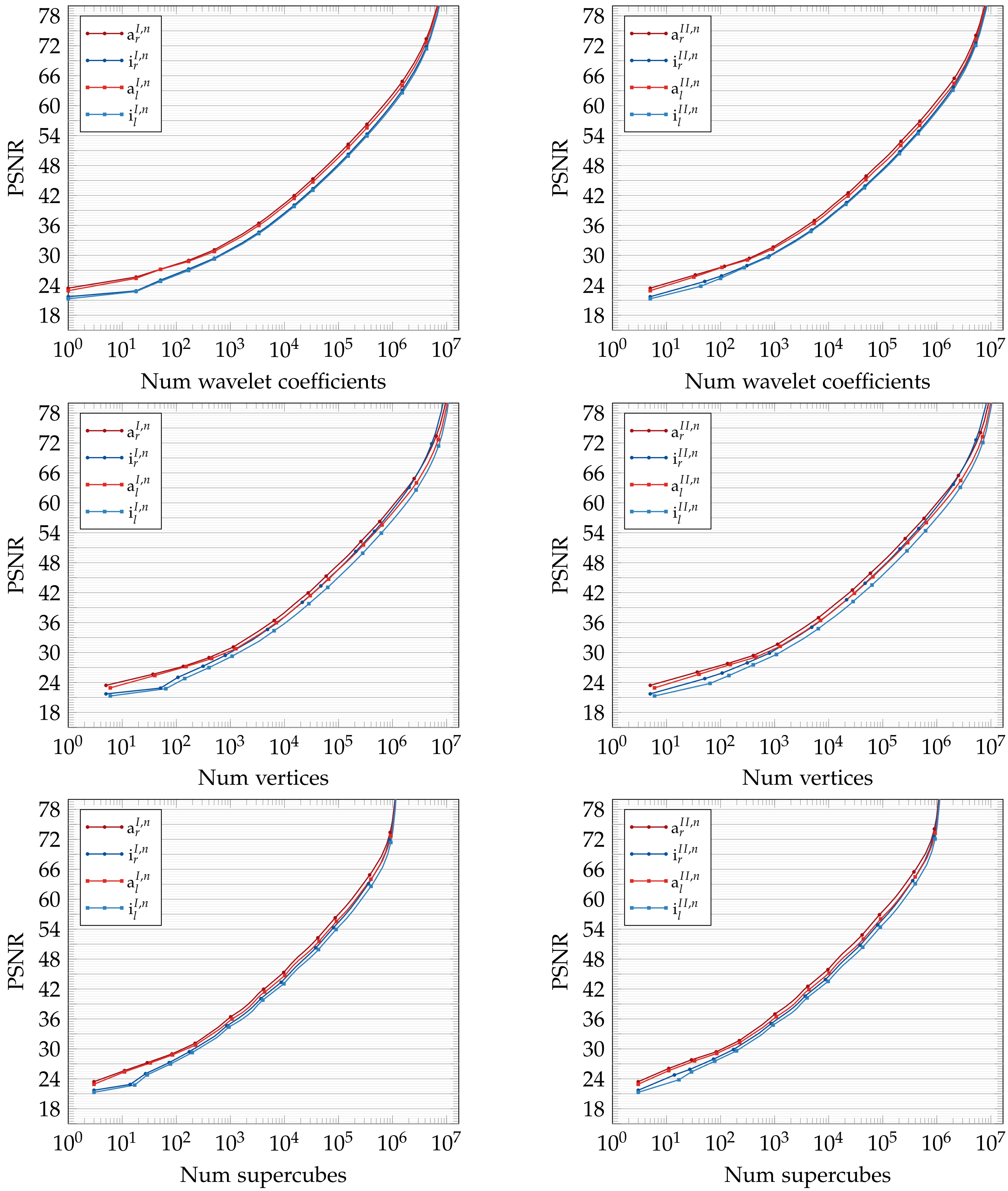

Figure 4.4: Puget 4K (4097^2) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Part II

3D

Chapter 5

Fuel (65^3)

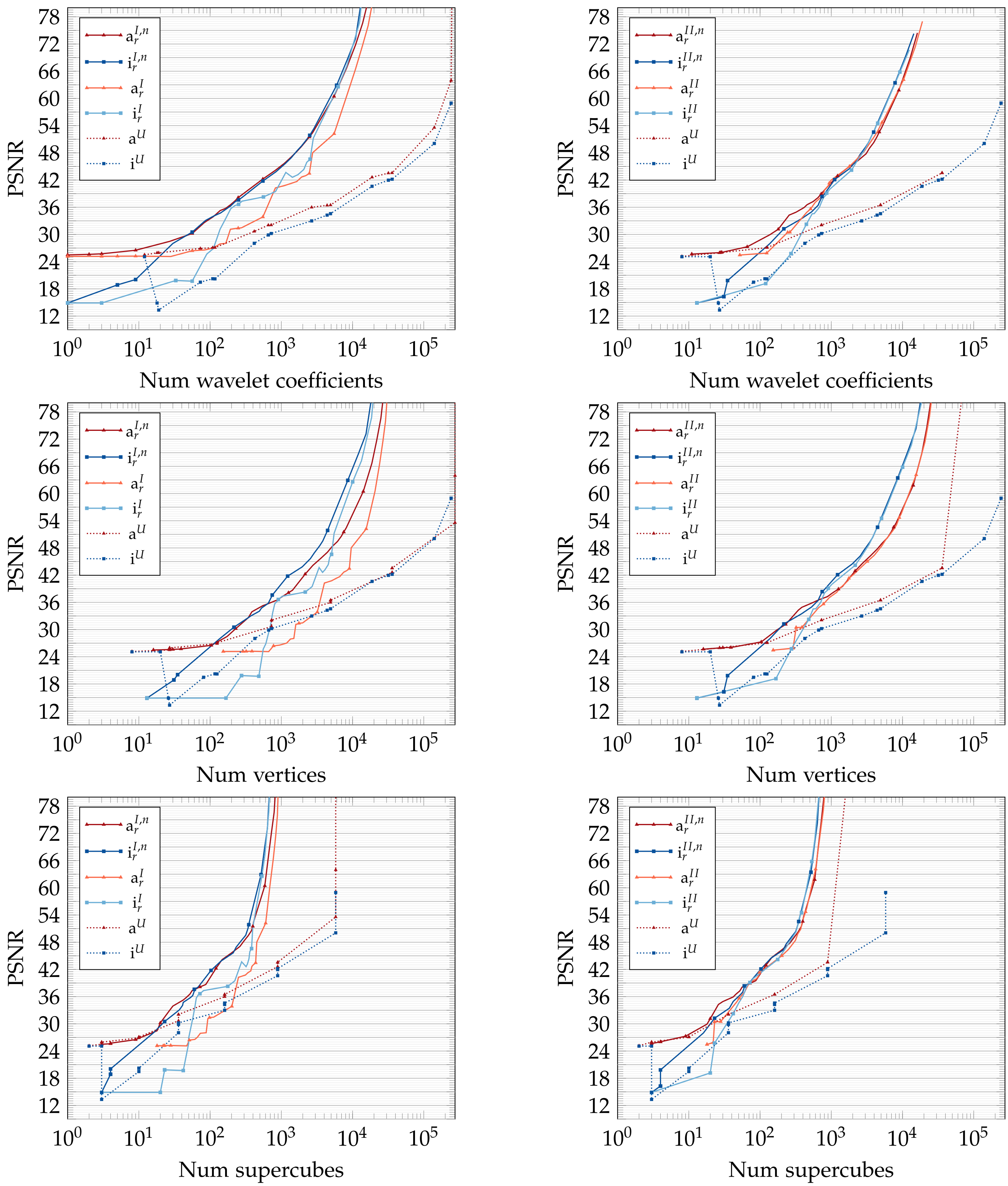

Figure 5.1: Fuel (65^3) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

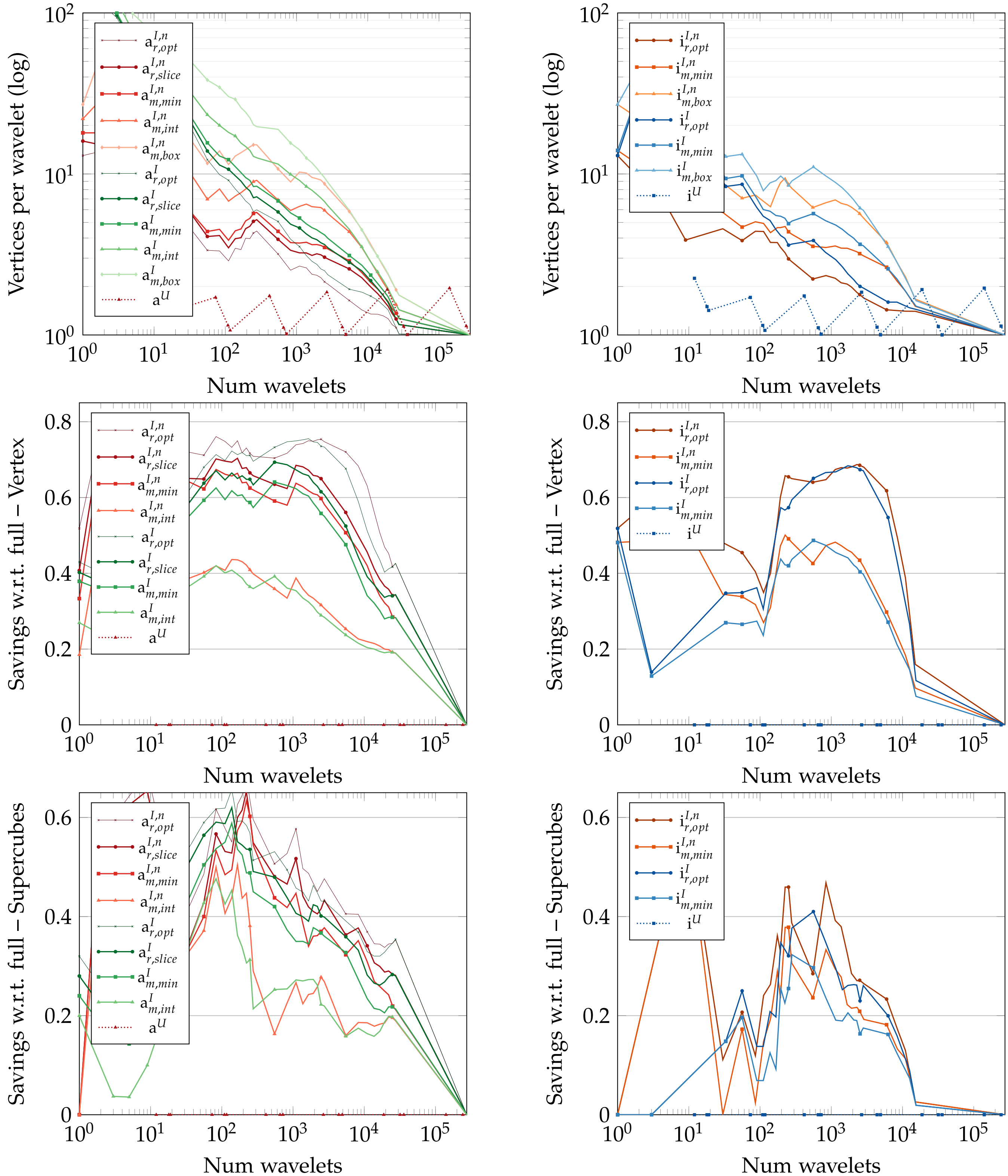

Figure 5.2: Fuel (65^3) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil}/vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil}/super_{fullGrid})$.

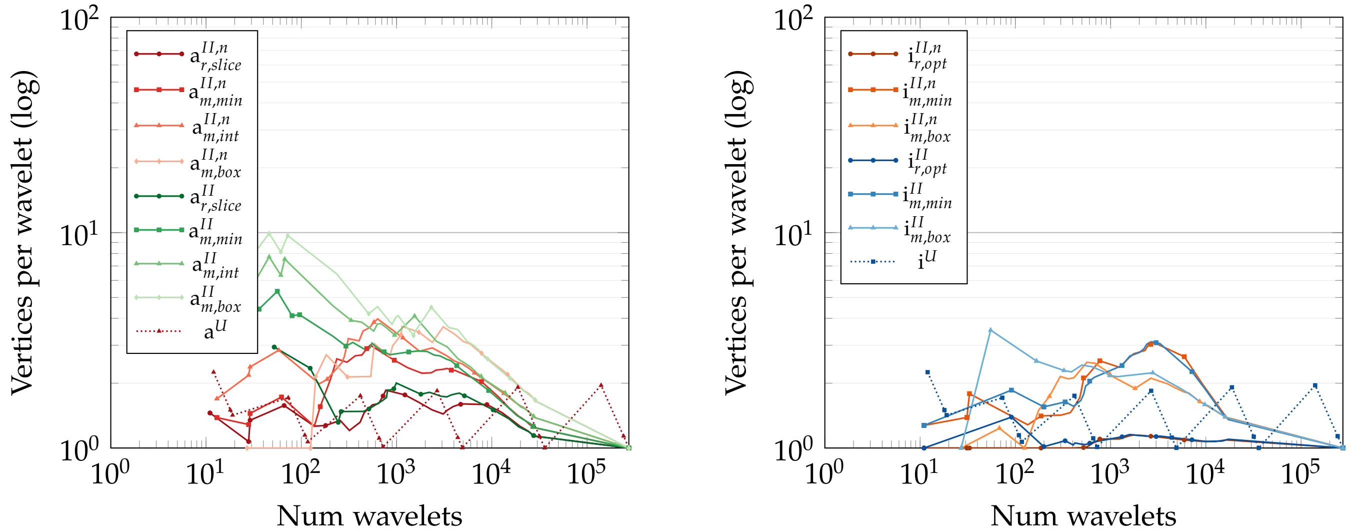

Figure 5.3: Fuel (65^3) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

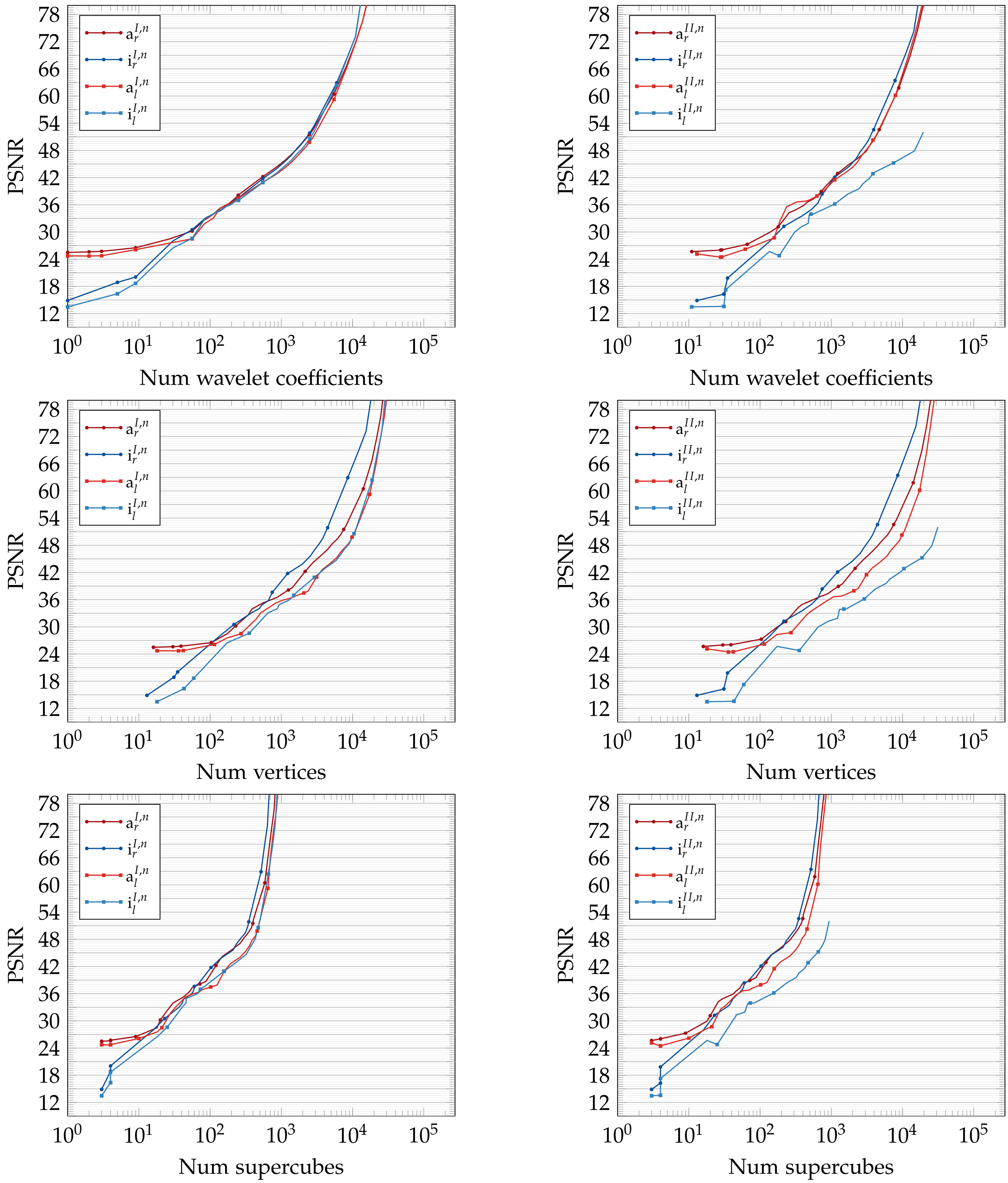

Figure 5.4: Fuel (65^3) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 6

Hydrogen (129^3)

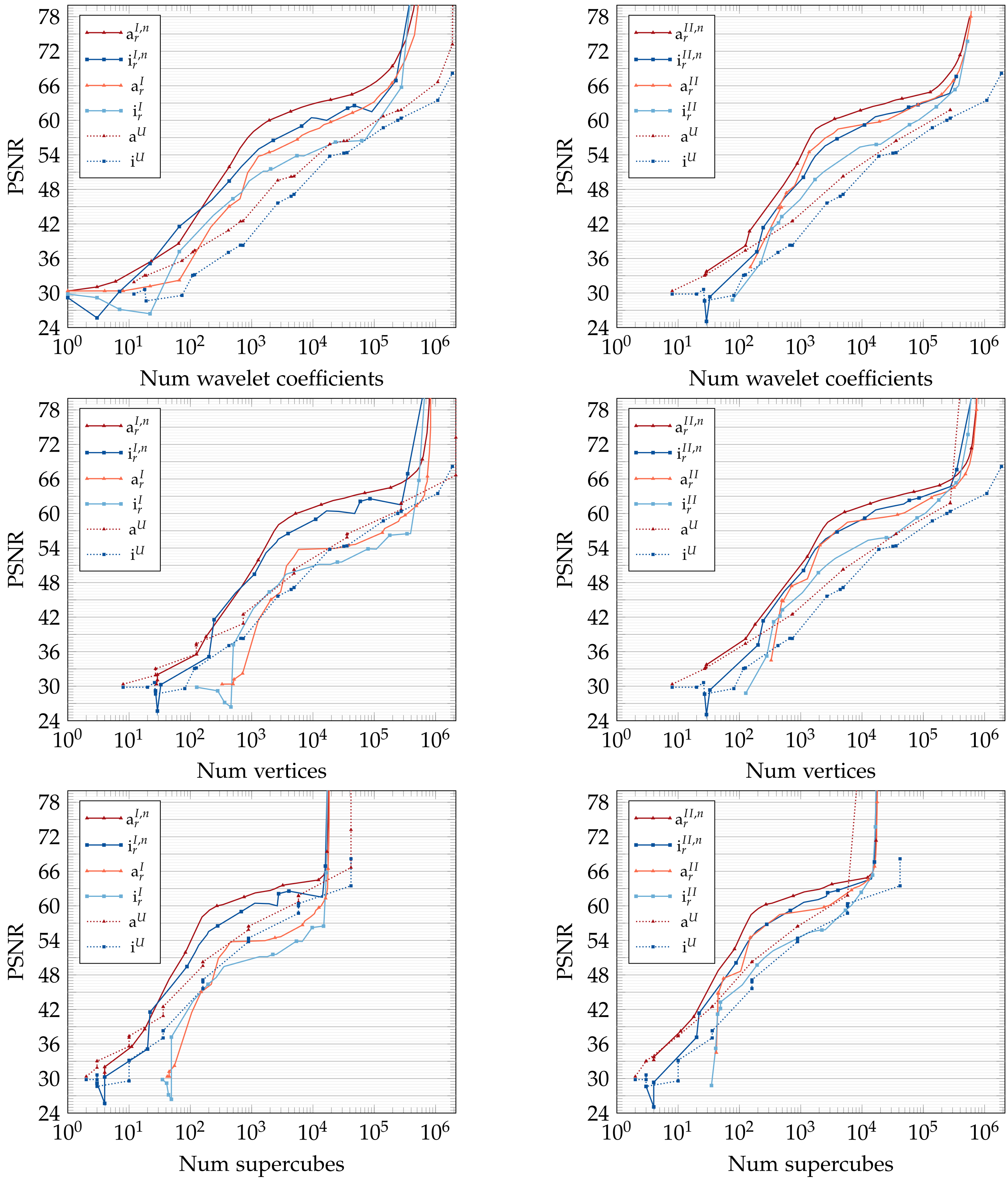

Figure 6.1: Hydrogen (129^3) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

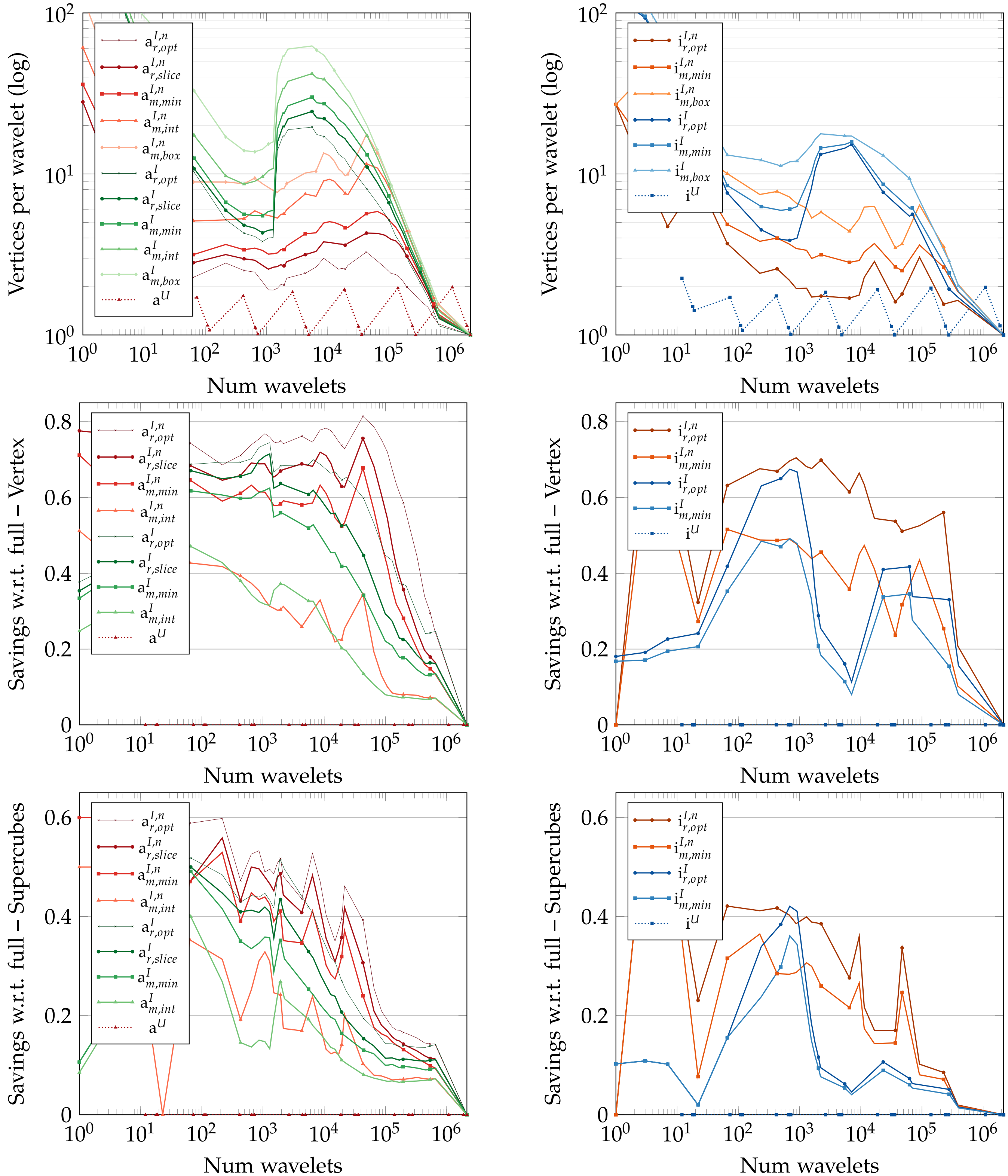

Figure 6.2: Hydrogen (129^3) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil} / vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil} / super_{fullGrid})$.

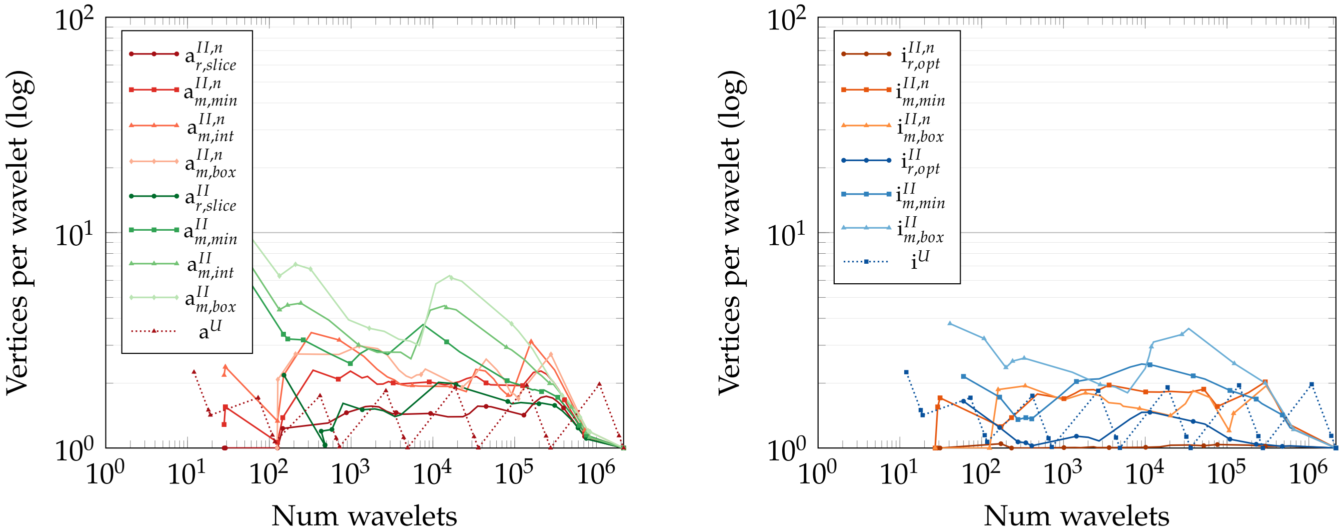

Figure 6.3: Hydrogen (129^3) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

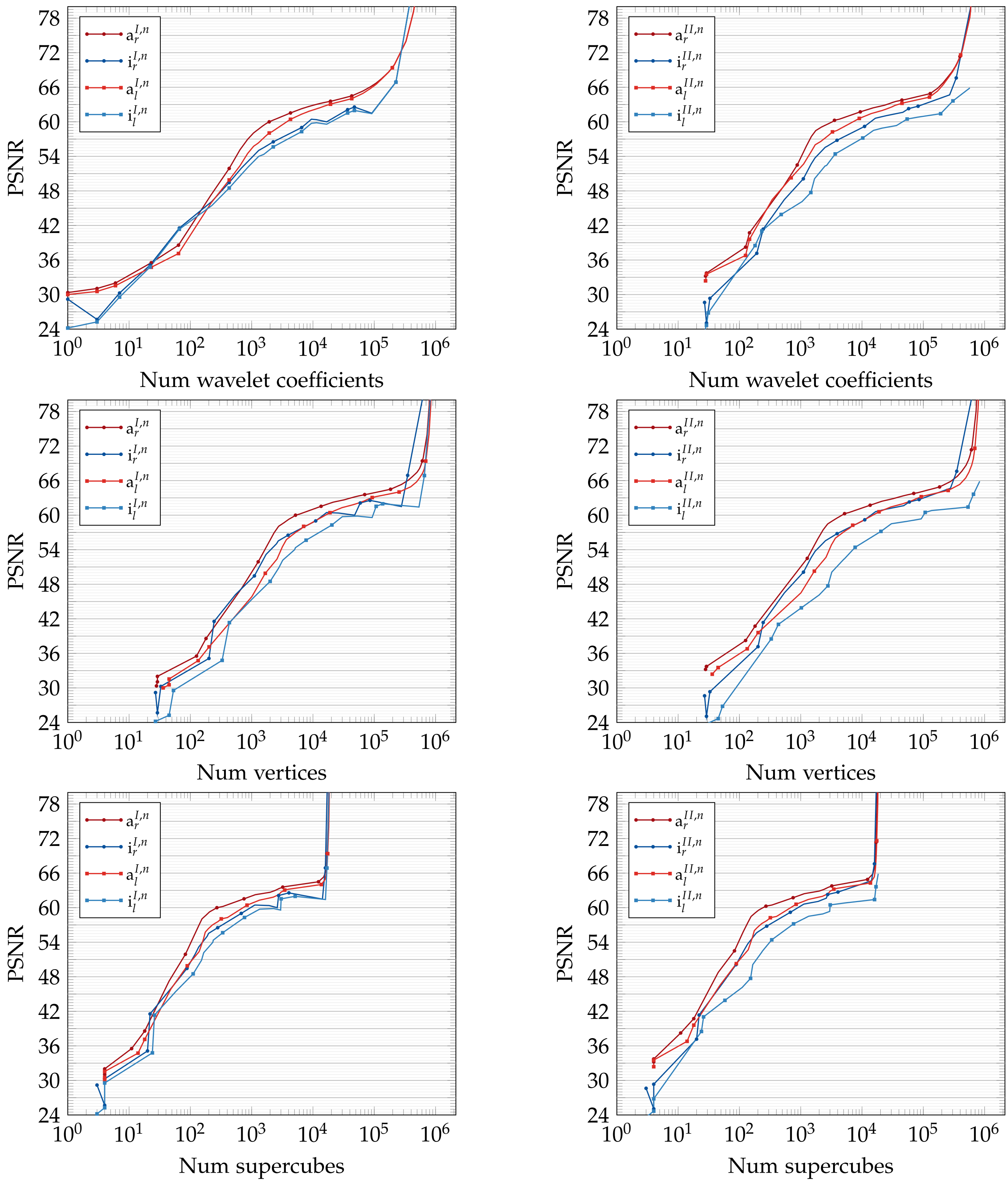

Figure 6.4: Hydrogen (129^3) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 7

Tooth (257^3)

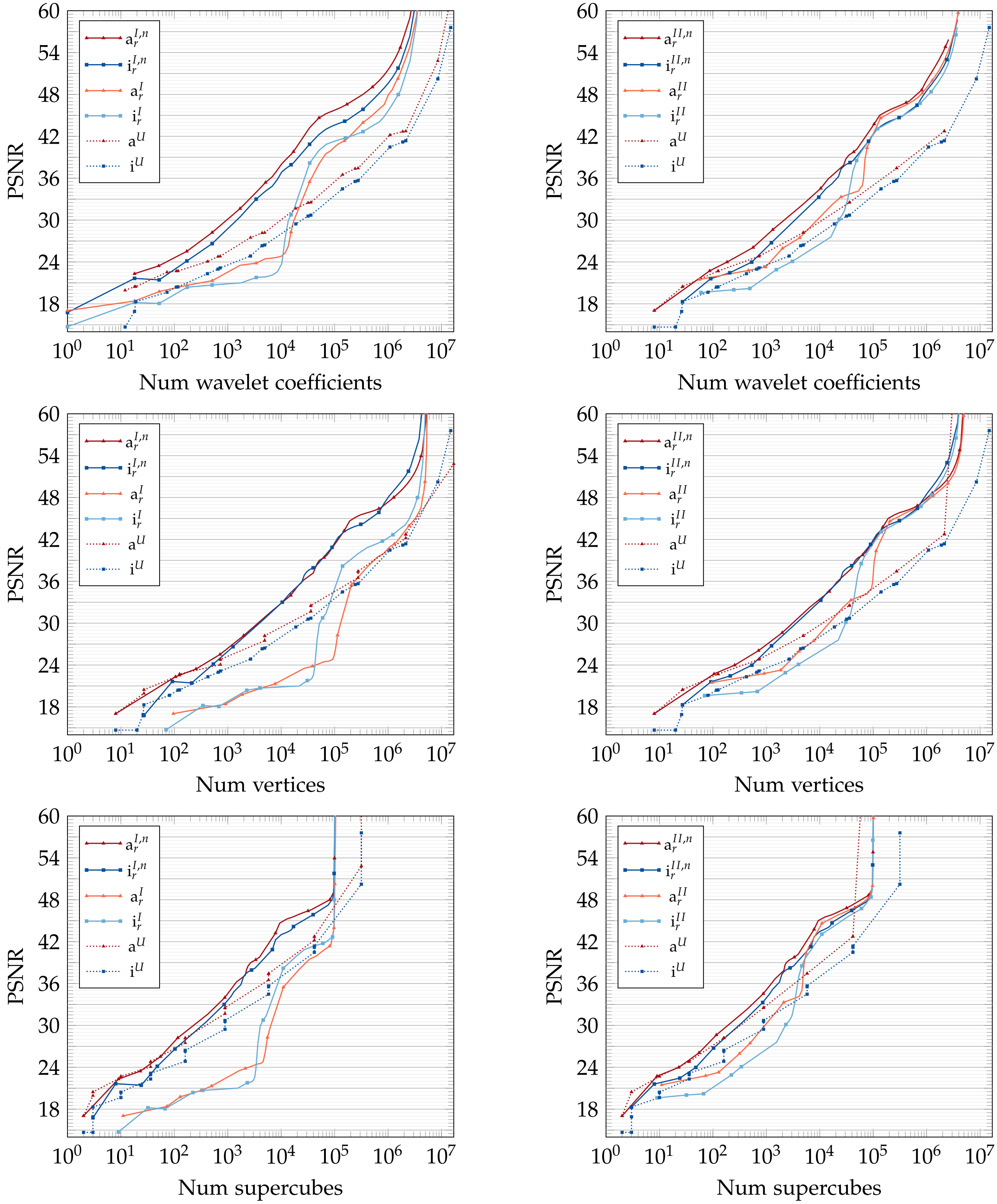

Figure 7.1: Tooth (257^3) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

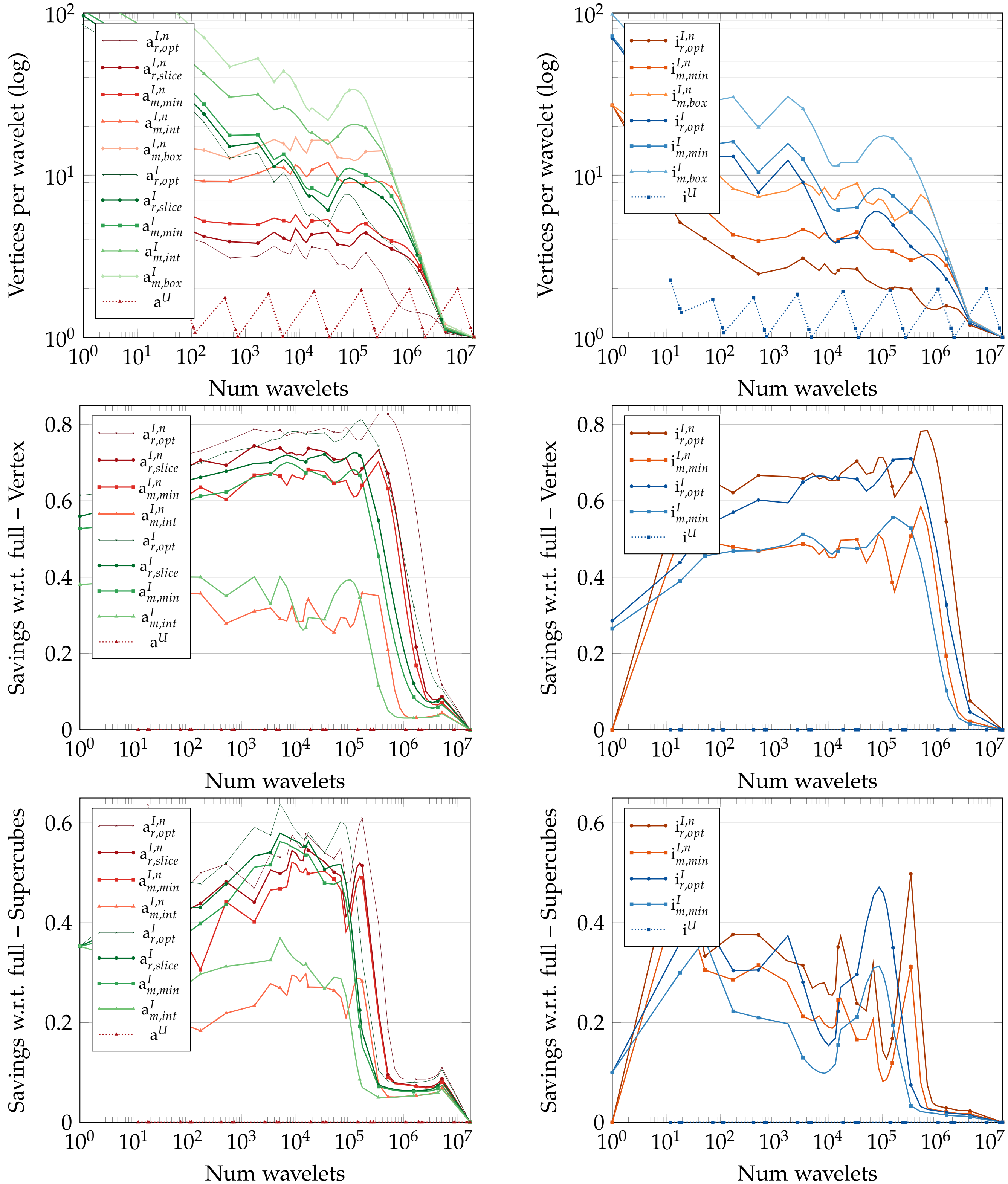

Figure 7.2: Tooth (257^3) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil}/vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil}/super_{fullGrid})$.

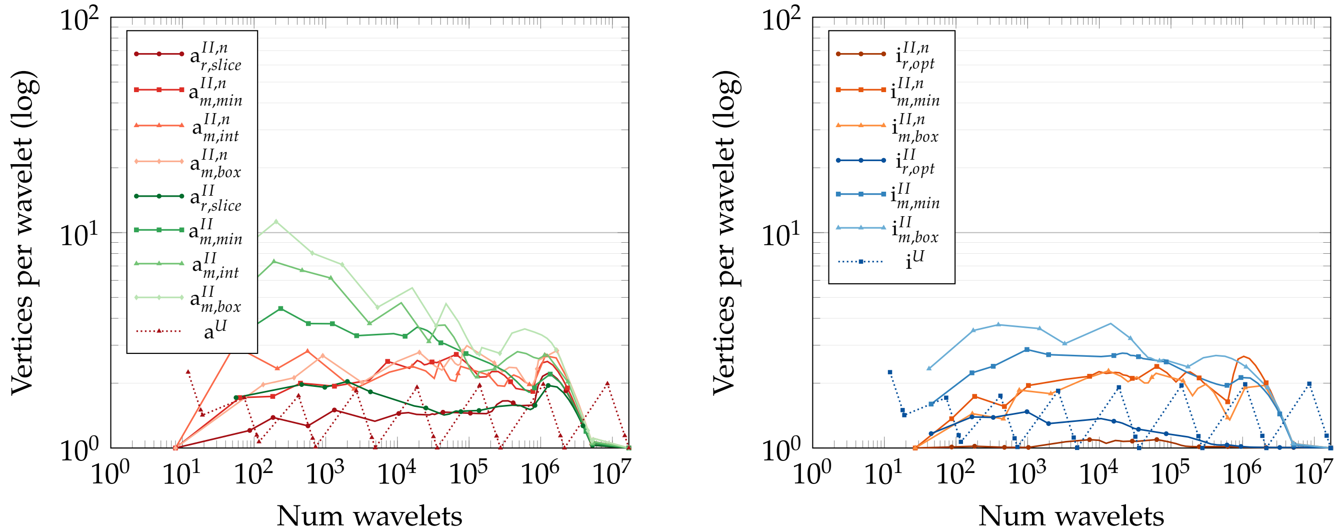

Figure 7.3: Tooth (257^3) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

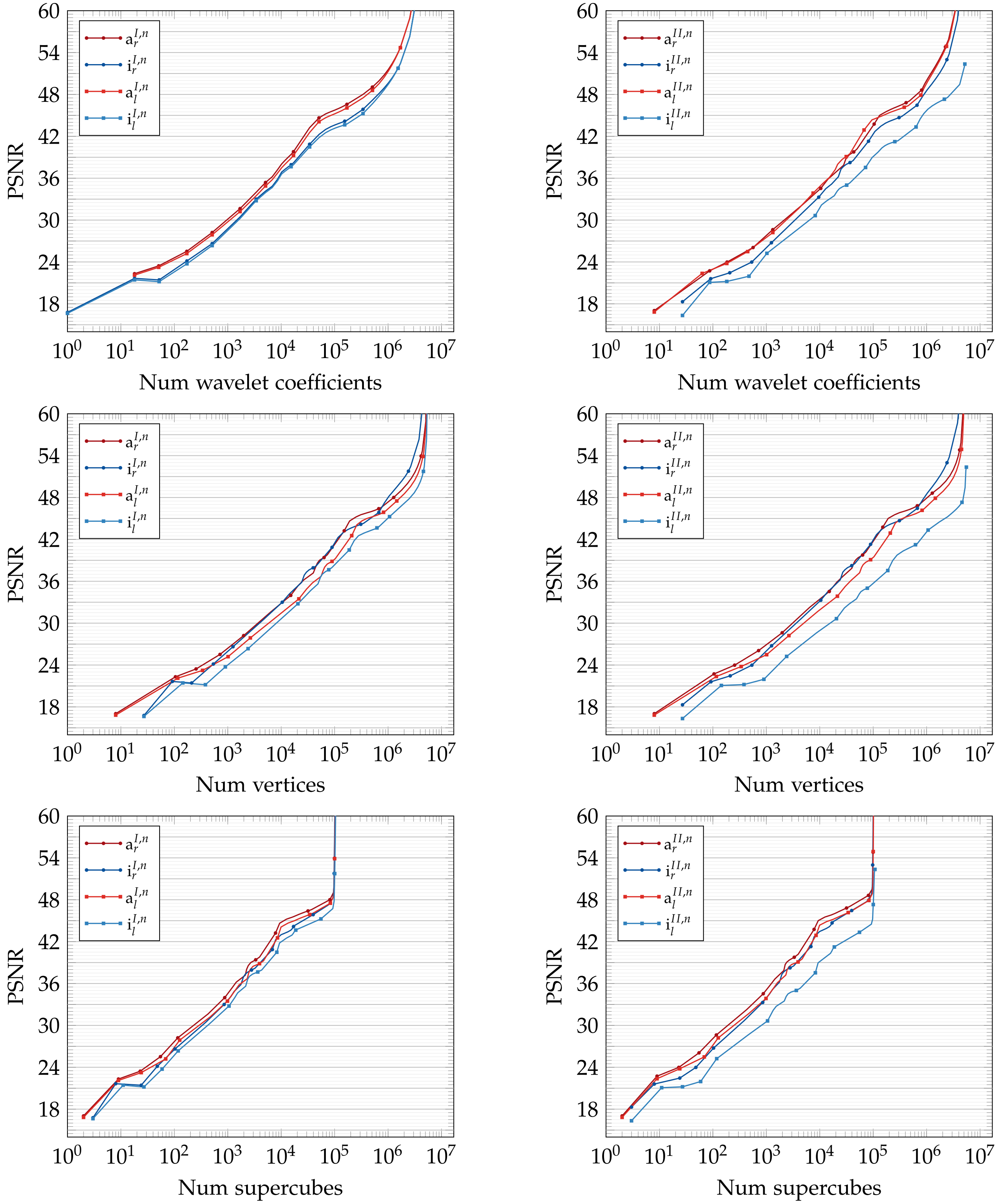

Figure 7.4: Tooth (257^3) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).

Chapter 8

Rayleigh-Taylor Instability (RTI) (257^3)

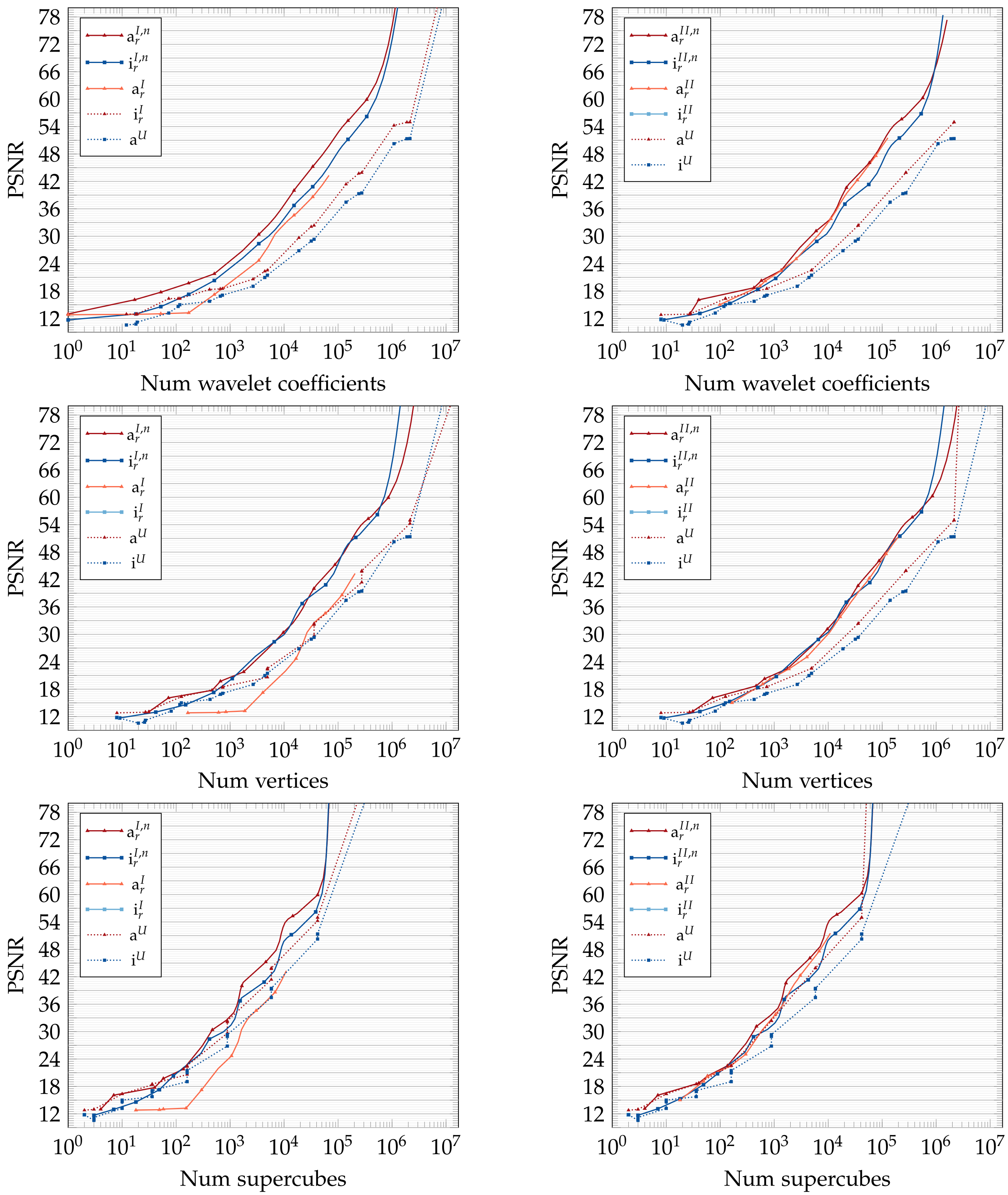

Figure 8.1: Rayleigh-Taylor Instability (RTI) (257^3) – Comparing the number of wavelets (TOP), vertices (MID) and supercubes (BOT) against PSNR, for unsaturated (Type I, left) and saturated (Type II, right) meshes. Each chart compares the {normalized and unnormalized} adaptive {approximating and interpolating} meshes as well as the uniform {approximating and interpolating} meshes at each depth in the hierarchy (all using the radial interpolant). These are plotted with dotted lines since the mesh only exists at the discrete sampled depths. In contrast, the adaptive datasets can be extracted at a virtually continuous level of detail (i.e. for any given tolerance). In all charts, vertical lines enable comparisons of PSNR across a fixed number of wavelets, vertices or supercubes, while horizontal lines enable comparisons of the number of wavelets, vertices or supercubes necessary to achieve a given PSNR.

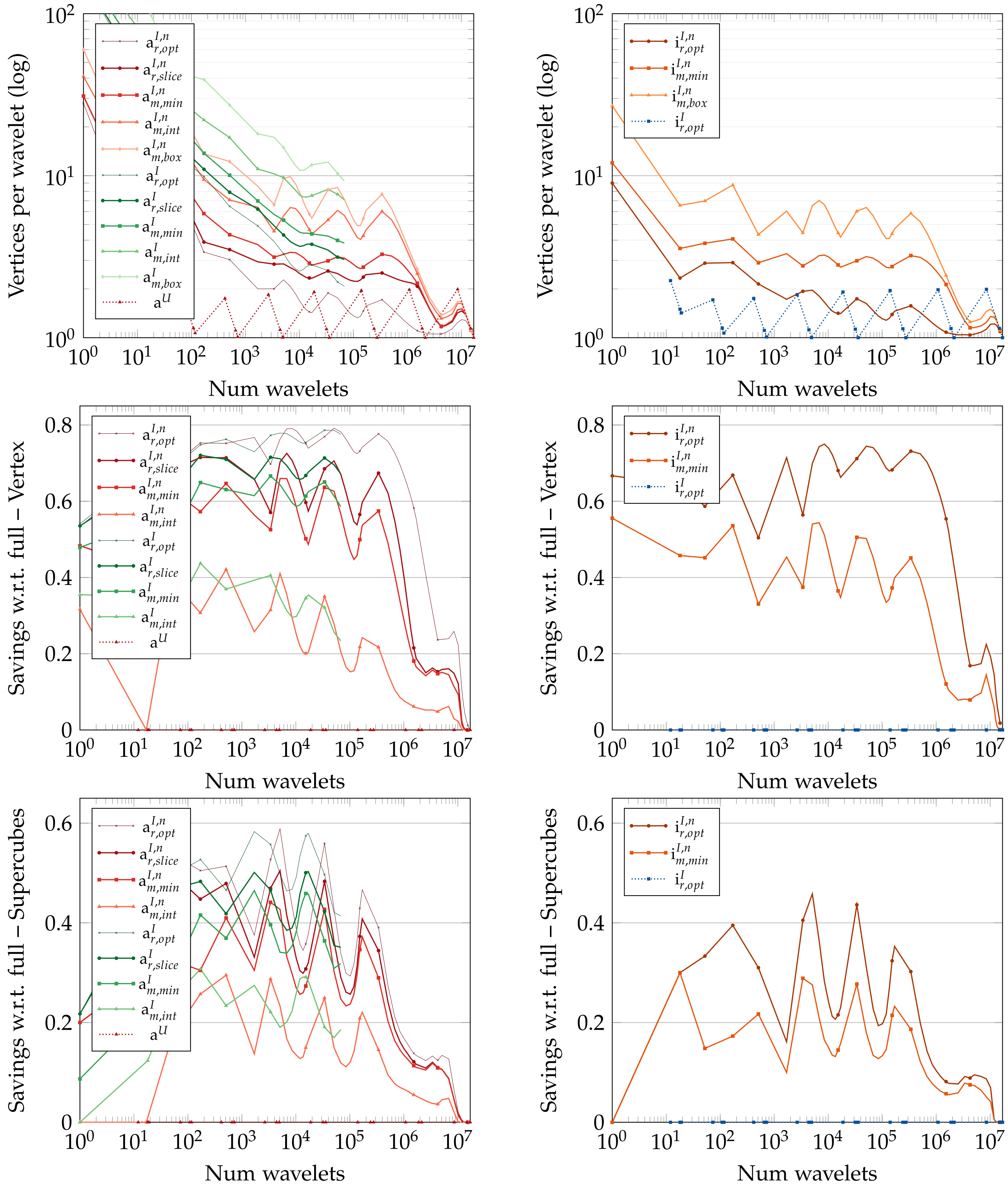

Figure 8.2: Rayleigh-Taylor Instability (RTI) (257^3) – Overhead of the different stencils for unsaturated (Type I) approximating (left) and interpolating (right) wavelets. Vertical lines correspond to meshes extracted using the same number of wavelets. For approximating wavelets, the thin lines (left) are the optimal (but incorrect) smallest radially interpolated stencil ($a^{\cdot}_{r,opt}$). Note that the conservative radial stencil ($a^{\cdot}_{r,slice}$) is typically pretty close to optimal. In all charts, the x-axis is the number of (Type I) wavelets, while the y-axis is: [TOP] (vertices per wavelet), where the mesh is generated according to a given stencil. [MID] savings (in vertices) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (vert_{currentStencil}/vert_{fullGrid})$. [BOT] savings (in supercubes) achieved by current stencil, compared to original subgrid stencil ($a^{I,*}_{m,box}$) – i.e. $1 - (super_{currentStencil}/super_{fullGrid})$.

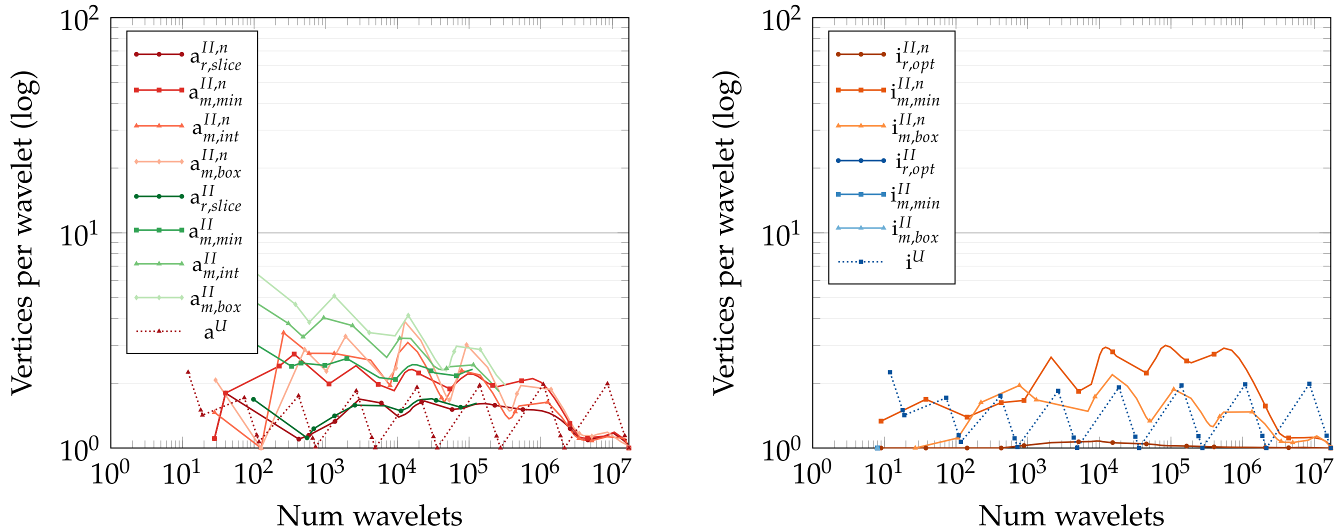

Figure 8.3: Rayleigh-Taylor Instability (RTI) (257^3) – Number of vertices per saturated (Type II) wavelet for the various stencil schemes. This uses the same meshes as the unsaturated (Type I) schemes on the top of the previous page. Note that for the normalized radial meshes, nearly all vertices have the proper support to be saturated.

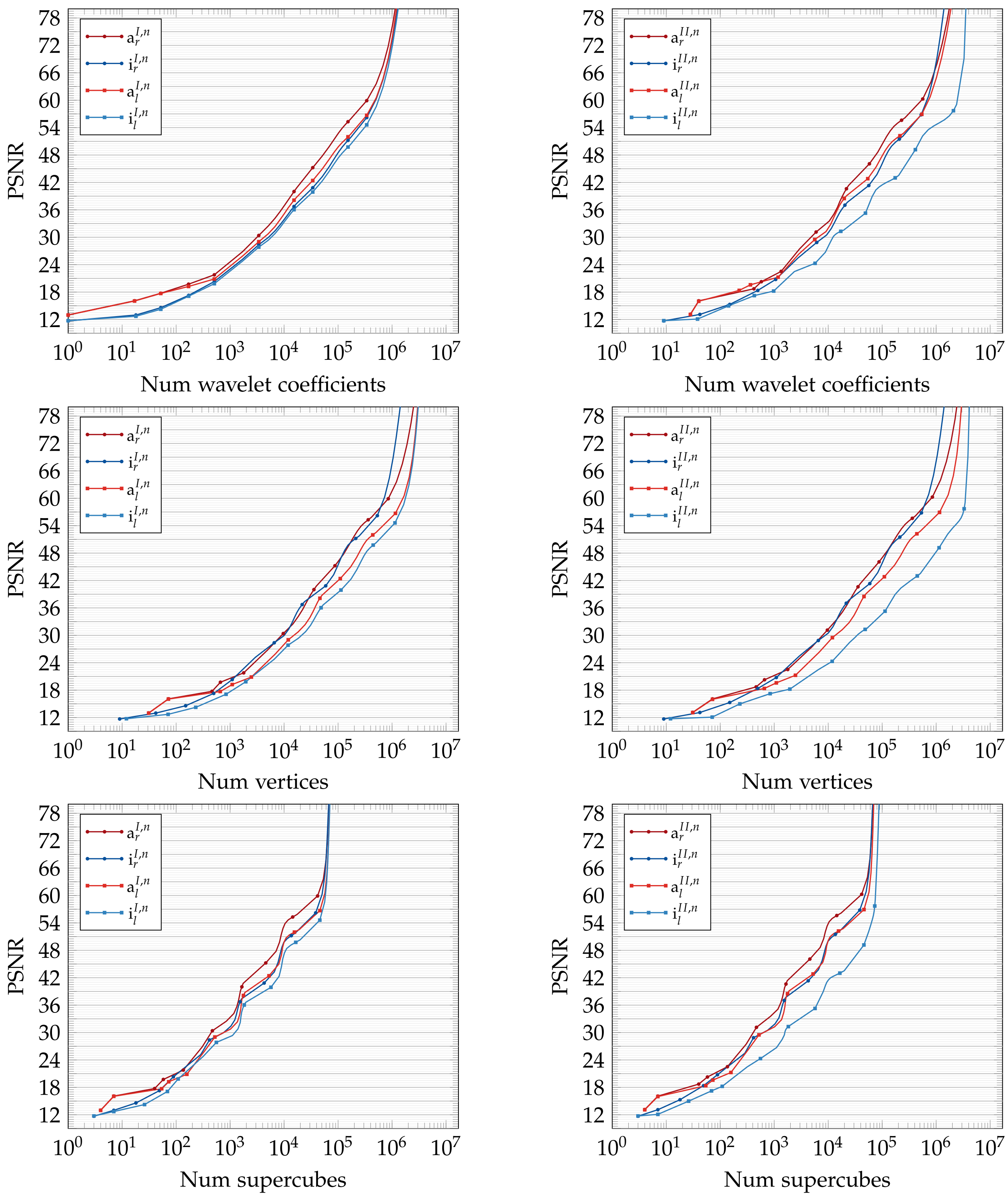

Figure 8.4: Rayleigh-Taylor Instability (RTI) (257^3) – Comparison of different interpolation schemes (darker: multilinear interpolant over cubes, lighter: linear interpolant over a triangulated domain) for approximating (red) and interpolating (blue) linear B-spline tensor product wavelets. Rows show: Number of wavelets (TOP), vertices (MID) and supercubes (BOT) vs. PSNR – using wavelet unsaturated (Type I, left) and saturated (Type II,right).