Differentiable Laplacian Matrix Guided Superpixel Segmentation

Supplementary Material

6. Class Count Plots of BSDS 500 Square Grid

The weighted reconstruction loss used a weight of 1.0 for pixels in *mixed-blocks* and 0.10 for pixels in *single-class blocks* as described in Sec. 3.3. Figure 7 shows the square-block grid on an example image and its corresponding label. As shown, most blocks contain only a single semantic class. Across the entire image, 61% contain exactly one class. The statistics for the BSDS500 training and validation splits are reported in Fig. 8. In both splits, more than 70% of blocks are single-class, providing direct motivation for assigning a higher per-pixel weight (10:1) to pixels in mixed-class regions.

7. Reconstructing $\mathbf{y}'_i$ from Q

Recall from Sec. 3.3 that the (weighted) reconstruction loss requires a reconstructed pixel property $\mathbf{y}'_i$. Here, we detail the steps that map pixel-wise properties into superpixel space and project them back via the soft assignment matrix Q, as widely used in the literature [16, 17, 43, 48, 52].

Superpixel aggregation: For each superpixel s, we compute a representative property vector $\mathbf{y}_s$ as the Q-weighted average of the ground-truth properties $\mathbf{y}_i$ of all pixels i that can be assigned to s (i.e., those with $i \in \mathcal{P}_s$), thereby aggregating local pixel information into a single representative property per superpixel:

$$\mathbf{y}_s = \frac{\sum_{i \in \mathcal{P}_s} \mathbf{y}_i \, q_{i,s}}{\sum_{i \in \mathcal{P}_s} q_{i,s}}. \tag{13}$$

Pixel-level reconstruction: The property $\mathbf{y}'_i$ at each pixel i is reconstructed by distributing the superpixel-level property values back to the pixel according to the pixel's assignment probabilities:

$$\mathbf{y}'_i = \sum_{s \in \mathcal{S}_i} \mathbf{y}_s \, q_{i,s}. \tag{14}$$

In summary, Q is used twice: first to pool pixel labels into superpixels, and then to redistribute these superpixel labels back to pixel space. The resulting reconstructed $\mathbf{y}'_i$ is compared to the ground-truth property $\mathbf{y}_i$ within the reconstruction loss.

8. Additional Results

The visual results on the Pascal VOC 2012 dataset follow the same patterns observed on BSDS500 and NYUv2. As shown in the EC standards results Fig. 9, the no-EC results Fig. 10, and our custom metrics Fig. 11:

- ASA and BR are largely unchanged through the incorporation of the Laplacian loss $\mathcal{L}_{\text{LAP}}$.
- Superpixel quality metrics improve, including compactness (CO), average cross-class consistency XC_{avg}, average stray-pixel rate ST_{avg}.

The AUC quantitative metrics for NYUv2 and Pascal VOC 2012 are provided in Tab. 4 and Tab. 5. Consistent with BSDS500, models using $\mathcal{L}_{\text{LAP}}$ exhibit comparable ASA but but improved CO_{AUC}, B_{AUC}, XC_{AUC}, and ST_{avg}. Visual examples in Fig. 12 further highlight the improved superpixel coherence and structure with Laplacian regularization.

Table 4. Quantitative performance on the inference-only **NYUv2** dataset. All results use 430–2023 superpixels. The best value in each EC setting is bolded. LAP models consistently have higher B_{AUC} and CO_{AUC} and lower XC_{AUC} and ST_{AUC}.

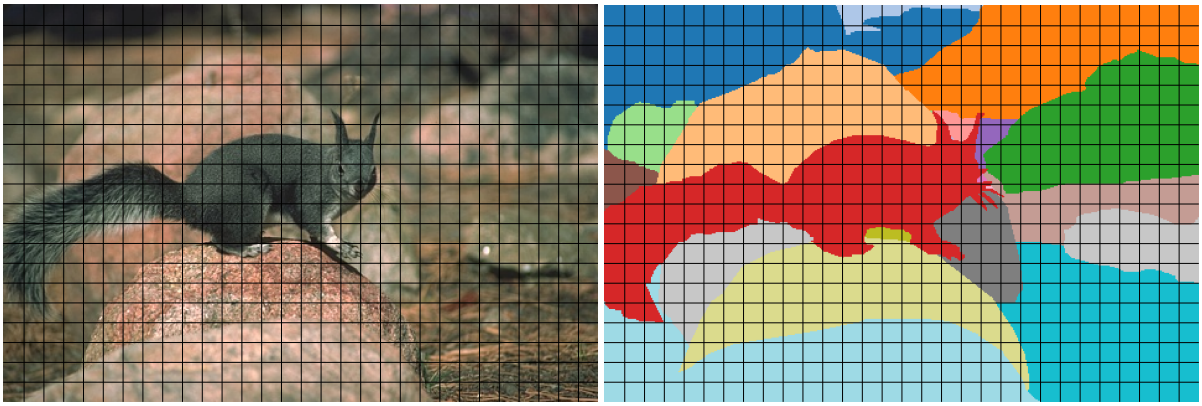
	Model	$\text{ASA}_{\text{AUC}}\uparrow$	$\text{CO}_{\text{AUC}}\uparrow$	$\text{B}_{\text{AUC}}\uparrow$	$XC_{\text{AUC}}\downarrow$	$\text{ST}_{\text{AUC}}\downarrow$
With EC	SCN	0.9443	**0.3791**	0.1879	0.00	0.00
	AINET	0.9429	0.3602	0.1857	0.00	0.00
	CDS	0.9460	0.3712	0.1895	0.00	0.00
	SSM	**0.9483**	0.3692	**0.1936**	0.00	0.00
	SCN-LAP	0.9406	**0.4725**	0.1955	0.00	0.00
	AINET-LAP	0.9425	0.4713	0.1938	0.00	0.00
	CDS-LAP	0.9459	0.4387	**0.2031**	0.00	0.00
	SSM-LAP	**0.9471**	0.4287	0.2015	0.00	0.00
Without EC	SCN	0.9456	**0.3651**	0.1837	290.04	1546.25
	AINET	0.9441	0.3415	0.1812	455.53	2314.21
	CDS	0.9472	0.3607	0.1852	292.19	**1417.75**
	SSM	**0.9496**	0.3590	**0.1894**	**265.43**	1441.60
	SCN-LAP	0.9420	0.4713	0.1925	169.68	1213.77
	AINET-LAP	0.9426	**0.4749**	0.1958	177.52	922.06
	CDS-LAP	**0.9486**	0.4439	0.1965	133.39	**773.06**
	SSM-LAP	0.9485	0.4294	**0.1985**	**132.64**	930.47

Table 5. Quantitative performance on **Pascal VOC 2012**, inference-only dataset. All results use 220–1300 superpixels. The best value in each EC setting is bolded. This third data follows the same results as the BSDS500 and NYUv2 dataset.

	Model	$\text{ASA}_{\text{AUC}}\uparrow$	$\text{CO}_{\text{AUC}}\uparrow$	$\text{B}_{\text{AUC}}\uparrow$	$XC_{\text{AUC}}\downarrow$	$\text{ST}_{\text{AUC}}\downarrow$
With EC	SCN	0.9818	**0.3671**	0.0632	0.00	0.00
	AINET	0.9821	0.3558	0.0625	0.00	0.00
	CDS	0.9825	0.3601	0.0651	0.00	0.00
	SSM	**0.9835**	0.3626	**0.0646**	0.00	0.00
	SCN-LAP	0.9817	**0.4646**	0.0664	0.00	0.00
	AINET-LAP	0.9814	0.4562	0.0697	0.00	0.00
	CDS-LAP	0.9830	0.4341	**0.0698**	0.00	0.00
	SSM-LAP	**0.9832**	0.4191	0.0679	0.00	0.00
Without EC	SCN	0.9827	**0.3496**	0.0614	**215.63**	**1181.76**
	AINET	0.9827	0.3261	0.0615	359.77	1851.90
	CDS	0.9835	0.3422	0.0629	255.41	1327.55
	SSM	**0.9842**	0.3418	**0.0639**	248.13	1447.70
	SCN-LAP	0.9821	**0.4570**	0.0665	**132.24**	927.87
	AINET-LAP	0.9821	0.4497	0.0698	140.46	**733.77**
	CDS-LAP	0.9837	0.4266	**0.0700**	138.46	813.91
	SSM-LAP	**0.9839**	0.4105	0.0679	142.89	998.36

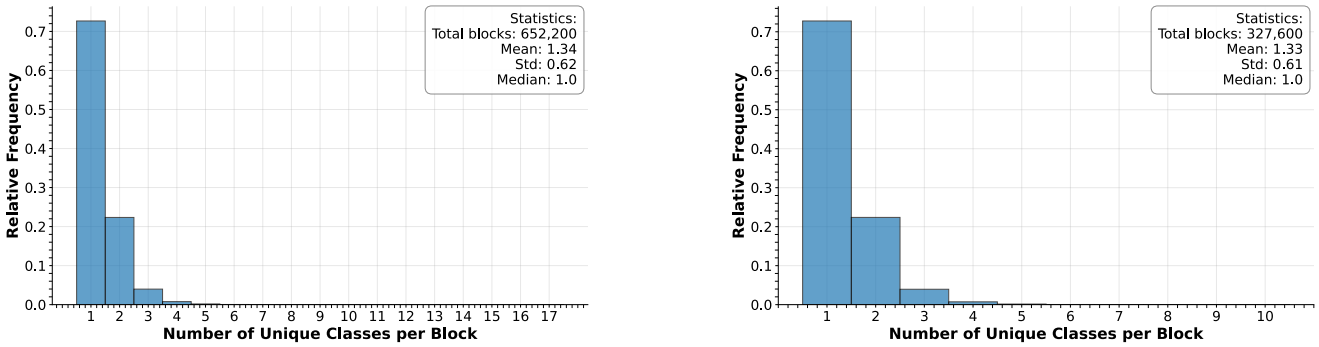
9. Area Under the Curve Metrics

Below we provide the implementation details for all area-under-the-curve (AUC) metrics introduced in Sec. 4.2. For



(a) Original image (b) Label

Figure 7. This figure shows what the square grid of blocks looks like on the original image (a), and one label(b). (b) Contains blocks which have 5 classes however 61% of blocks contain a single class. This illustrates that the image complexity in block perspective resides in a small portion of the entire image. This motivates the use of the weighted reconstruction loss Sec. 3.3.



(a) Training Set (b) Validation Set

Figure 8. Distribution of class diversity within superpixel grid initialization blocks across the train and validation splits of BSDS500. The histograms display the relative frequency of blocks containing different numbers of unique classes. Recall that the block size is 16×16 pixels, $|\mathcal{B}|$ which means 600 superpixels for the original image sizes (320×420) in BSDS500. (a) Shows the class diversity distribution for the training set and (b) the validation set. The splits are consistent with about 90% of blocks containing either 1 or 2 classes. While 70% of blocks contain only a single class.

a fixed superpixel-count range $[n_{\min}, n_{\max}]$, each metric is normalized by the width of the range:

$$\mathrm{ASA}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{ASA}(n)\, dn, \tag{15}$$

$$\mathrm{CO}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{CO}(n)\, dn, \tag{16}$$

$$\mathrm{BR}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{BR}(n)\, dn, \tag{17}$$

$$\mathrm{BP}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{BP}(n)\, dn, \tag{18}$$

$$\mathrm{ST}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{Stray}_{\mathrm{avg}}(n)\, dn, \tag{19}$$

$$\mathrm{B}_{\mathrm{AUC}} = \mathrm{BR}_{\mathrm{AUC}} \times \mathrm{BP}_{\mathrm{AUC}}, \tag{20}$$

$$\mathrm{XC}_{\mathrm{AUC}} = \frac{1}{n_{\max} - n_{\min}} \int_{n_{\min}}^{n_{\max}} \mathrm{XC}_{\mathrm{avg}}(n)\, dn. \tag{21}$$

10. Note on Superpixel Graph Construction

Fig. 13 visualizes the graph construction from the perspective of one pixel and shows a high-level flow diagram of the Laplacian loss. The definition of a node degree in Eq. (1) assumes that each pixel has eight neighbors. While this holds for interior pixels, boundary pixels may have neighbors outside the image or outside the superpixel's pixel set $\mathcal{P}_s$, for superpixel s, due to the finite 3×3 block window. In our construction, such neighbors do not contribute to the graph $\mathcal{G}_s$. For a fixed superpixel s and a pixel $i \in \mathcal{P}_s$, which has a neighbor $j \in \mathcal{N}_i$ but lies outside $j \notin \mathcal{P}_s$, the corresponding

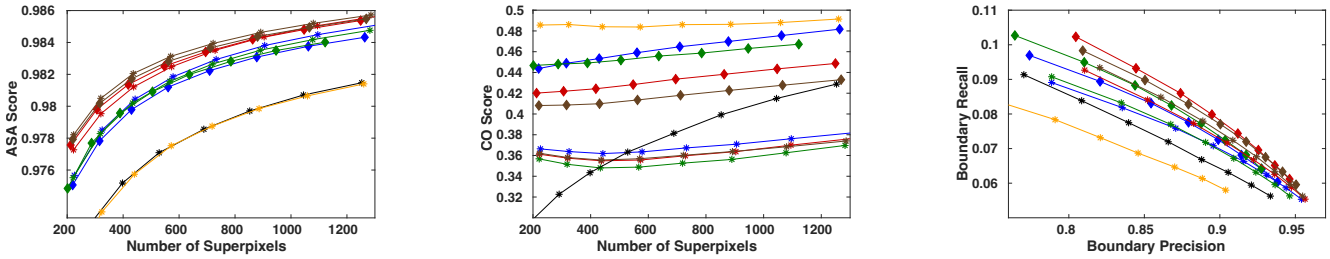


Figure 9. Standard metrics *with enforced connectivity (EC)* on Pascal VOC 2012. Columns (left→right): **ASA**, **CO**, and **BR–BP**. Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown), SIN(orange) and SLIC (black). Laplacian variants use the same color with a diamond (⋄) marker (not applicable to SLIC and SIN). Laplacian variant models out perform their counterparts on CO and BR with minimal impact on ASA and BP.

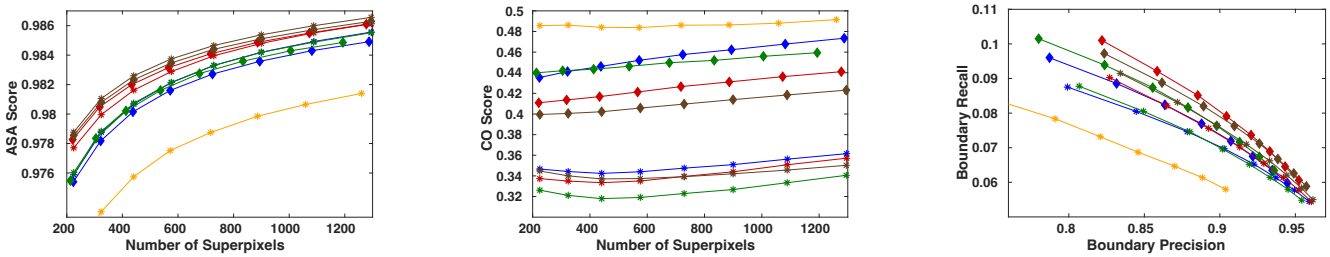


Figure 10. Standard metrics *without enforced connectivity (EC)* on Pascal VOC 2012. Columns (left→right): **ASA**, **CO**, and **BR–BP**. Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown), SIN (orange); Laplacian variants use the same color with a diamond (⋄) marker (not applicable to SIN). Laplacian variant models without EC out perform their counterparts on CO and BR with minimal impact on ASA and BP.

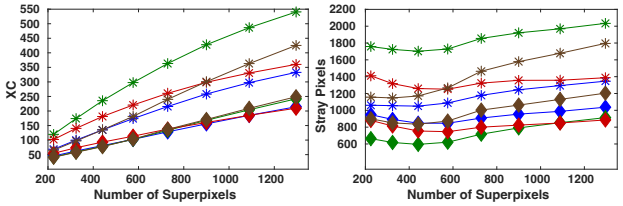


Figure 11. Fragmentation metrics *without enforced connectivity (EC)* on pascal VOC 2012. (left→right): average excess components counts (XC_{avg}) and average stray pixels counts ($\text{Stray}_{\text{avg}}$). Baselines are plotted with star markers—SCN (blue), AINet (green), CDS (red), SSM (brown). Laplacian variants use the same color with a diamond (⋄) marker lower metric values are less fragmented superpixels and all Laplacian variants are lower.

edge weight is set to zero, i.e.,

$$q_{i,s}q_{j,s} = 0,$$

and j is omitted from the graph $\mathcal{G}_s$. Consequently, the effective neighborhood of i in $\mathcal{G}_s$ is $\mathcal{N}_i \cap \mathcal{P}_s$, and the degree $d_{i,s}$ satisfies

$$d_{i,s} = \sum_{j \in \mathcal{N}_i \cap \mathcal{P}_s} q_{i,s}q_{j,s} \leq \left|\mathcal{N}_i \cap \mathcal{P}_s\right| \leq 8,$$

since $q_{i,s}, q_{j,s} \in [0,1]$ and each pixel has at most eight neighbors. Summing over all pixels $i \in \mathcal{P}_s$ gives:

$$\text{tr}(L_s) = \sum_{i \in \mathcal{P}_s} d_{i,s} \leq \sum_{i \in \mathcal{P}_s} \left|\mathcal{N}_i \cap \mathcal{P}_s\right| \leq 8N_s.$$

Thus, $8N_s$ is a valid global upper bound on the trace of the Laplacian for any graph $\mathcal{G}_s$ constructed from an 8-connected neighborhood.

We normalize the graph-Laplacian loss in Eq. (2) by this upper bound $8N_s$. This choice has two practical advantages.

1. It keeps $\text{tr}(L_s)/(8N_s)$ in $[0,1]$ with a simple, shape-independent normalization that does not require tracking the exact number of valid neighbors for each pixel.
2. It avoids introducing an explicit dependence of the normalization on the location of $i \in \mathcal{P}_s$, which would complicate the loss without changing the underlying connectivity objective.

In short, boundary superpixels inherently have fewer potential internal edges and therefore cannot achieve the same maximum trace as interior superpixels. Using a single global bound $8N_s$ for all superpixels provide a conservative yet consistent scaling for the graph-Laplacian, while the true connectivity structure of $\mathcal{G}_s$ remains determined solely by the actual pixel assignments within $\mathcal{P}_s$.

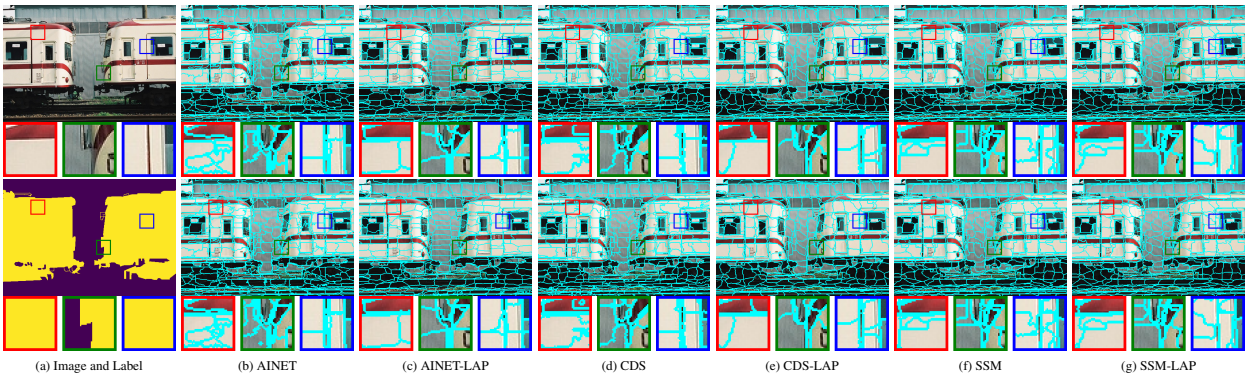


Figure 12. Qualitative comparison on **Pascal VOC 2012**. The first row shows the *input image* followed by outputs *with enforced connectivity (EC)* for: AINet, AINet-LAP, CDS, CDS-LAP, SSM, SSM-LAP. The second row shows the *ground-truth labels* followed by the corresponding outputs *without EC*. Colored boxes (red/green/blue) highlight regions where EC relabels fragmented components and where boundary irregularities are reduced; training with the proposed graph-Laplacian (LAP) loss yields more compact, connected superpixels and fewer label changes under EC.

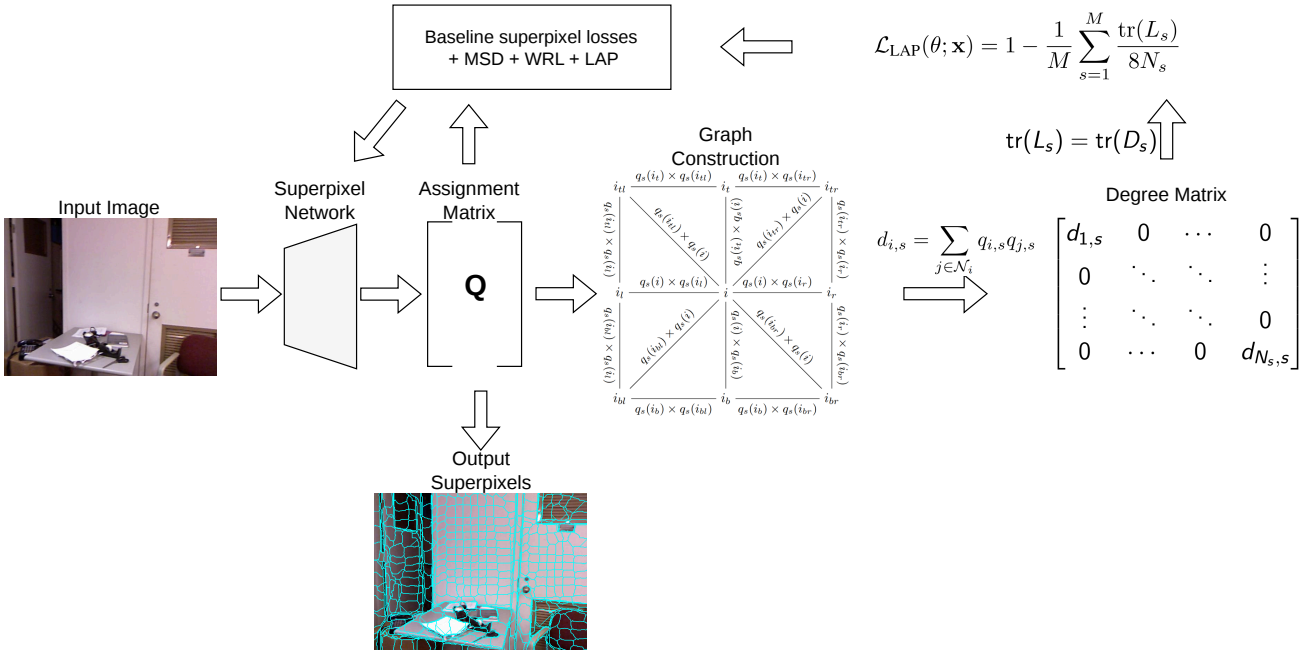


Figure 13. Overview of graph-Laplacian loss. The input image is passed to any superpixel network that outputs the assignment matrix $Q \in \mathbb{R}^{N\times M}$. From Q we construct a graph for each superpixel s and the corresponding degree matrix D_s. Since the graph has no self loops, $\text{tr}(L_s) = \text{tr}(D_s)$. Then we take the average Laplacian normalized by the maximum trace across all superpixels.

11. Trace Maximization as a Proxy for Connectivity

In this section, we provide insight into why trace maximization can serve as a proxy to improve superpixel connectivity.

Recall that $L_s \in \mathbb{R}^{N_s\times N_s}$ denotes the Laplacian of $\mathcal{G}_s$. Specifically, for $L_s = D_s - A_s$, we have:

$$L_s(i,j) = \begin{cases} d_{i,s} & \text{if } i = j, \\ -q_{i,s}q_{j,s} & \text{if } i \neq j, \\ 0 & \text{else.} \end{cases} \tag{22}$$

Let L_s eigenvalues be $0 = \lambda_{1,s} = \cdots = \lambda_{k_s,s} < \lambda_{k_s+1,s} \leq \cdots \leq \lambda_{N_s,s}$ where k_s is the geometric multiplicity of the null eigenvalue. It is well known that k_s equals the number of connected components in $\mathcal{G}_s$ [41]. Since L_s is symmetric

positive semi-definite, all eigenvalues are nonnegative and

$$\mathrm{tr}(L_s) = \sum_{i=1}^{N_s} \lambda_{i,s} = \sum_{i=k_s+1}^{N_s} \lambda_{i,s}. \tag{23}$$

The trace is equal to the sum of all strictly positive eigenvalues. Under our construction, each degree satisfies $d_{i,s} \leq 8$, which implies

$$\mathrm{tr}(L_s) = \sum_{i=1}^{N_s} d_{i,s} \leq 8N_s. \tag{24}$$

To determine an upperbound for the eigenvalues, we use Gershgorin Circle Theorem. According to the theorem, every eigenvalue λ of L_s lies inside at least one Gershgorin disc of radius

$$R_i = \sum_{j \neq i} |L_s(i,j)|. \tag{25}$$

This means that for every eigenvalue λ of L_s, we have $|\lambda - L_s(i,i)| \leq R_i$ for some i. Since the graph is undirected and weights are nonnegative, we have

$$R_i = \sum_{j \neq i} |L_s(i,j)| = \sum_{j \neq i} q_{i,s} q_{j,s} = d_{i,s}. \tag{26}$$

It follows that for every eigenvalue λ of L_s, we have

$$|\lambda - d_{i,s}| \leq d_{i,s}, \tag{27}$$

which implies

$$0 \leq \lambda \leq 2d_{i,s} \leq 2 \cdot 8 = 16. \tag{28}$$

Therefore, the trace is bounded as follows:

$$\mathrm{tr}(L_s) = \sum_{i=k_s+1}^{N_s} \lambda_{i,s} \leq 16(N_s - k_s). \tag{29}$$

This inequality makes the relationship between trace and connectivity explicit: for fixed N_s, increasing k_s (i.e., introducing more connected components) strictly decreases the maximum achievable trace. Conversely, achieving a large trace requires many positive eigenvalues, which necessarily implies small k_s.

In particular, the maximum trace is attained when $k_s = 1$, i.e., when the graph is fully connected, and all remaining $N_s - 1$ eigenvalues are as large as allowed under the degree constraints. While maximizing $\mathrm{tr}(L_s)$ does not algebraically force $k_s = 1$ in arbitrary graphs, in our bounded-degree, bounded-weight 8-neighborhood construction, the trace cannot be inflated by concentrating mass in a few extreme eigenvalues. Instead, a large trace requires widespread positive connectivity throughout the graph.

Hence, under the structural constraints of $\mathcal{G}_s$, maximizing

$$\frac{\mathrm{tr}(L_s)}{8N_s} \tag{30}$$

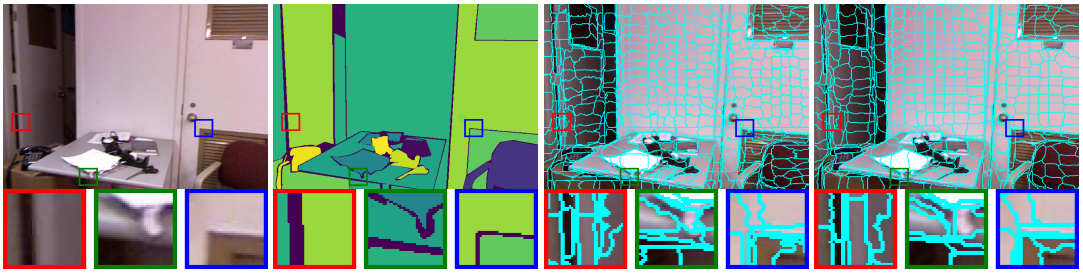
acts as a principled surrogate for minimizing the multiplicity k_s of the null eigenvalue, and therefore promotes spatial connectivity within each superpixel.

12. Model Failure Modes

The baselines SCN, AINet, CDSpixel, and SSM along with their LAP counterparts still suffer from an over segmentation of boundaries between classes. For example as shown in Fig. 14, the superpixels along such boundaries are irregular and smaller than superpixels further away from boundaries. As shown in Fig. 5, the LAP loss heavily encourages connected superpixels but fails to only produce fully connected superpixels.

13. Importance of Connectivity for Downstream Tasks

The goal of the LAP loss is for fully differentiable superpixel segmentation whose output is connected without hard decisions in post-processing. To demonstrate the importance of connectivity on downstream tasks, we followed SCN [52] and implemented our differentiable Laplacian loss on stereo matching, a computer vision task that estimates pixel-wise depth by finding correspondences between a pair of stereo images. Specifically, we conducted an experiment on the Monkaa subset of SceneFlow [29] where the superpixel and stereo matching networks are optimized jointly. Note that SCN turns off EC post-processing to allow for training with the task-specific network. Under the SCN [52] baseline, the end-point error (EPE) is 0.92, whereas integrating our Laplacian loss reduces the EPE to 0.88, a 4.3% improvement. This result confirms that improving connectivity alone allows superpixels to better align with downstream tasks. Despite deep-learning superpixel networks outperforming traditional algorithms on boundary recall and precision, traditional algorithms are still used in multiple downstream task applications due in part to the assumption of fully-connected compact superpixels [3, 7, 22, 25, 27, 30, 32, 35, 44, 50, 51, 58, 62, 63]. Future work involves incorporating LAP loss models into a variety of downstream tasks to explore if our improved superpixel connectivity and compactness is sufficient to meet downstream task requirements.



(a) Image (b) Label (c) CDS (d) CDS-LAP

Figure 14. Qualitative comparison on **NYUv2**. The figures shows the *input image*, the *ground-truth labels* followed by outputs *without enforced connectivity* for CDS and CDS-LAP. Colored boxes (red/green/blue) highlight regions where both models over segment class boundaries.