

1 Speeding up visualizations with WebGL

When visualizing large datasets with Bokeh, the interaction can become rather slow. To counter this, one can enable WebGL, which allows rendering some glyph types on graphics hardware.

1.1 What is WebGL?

WebGL is a JavaScript API that allows rendering content in the browser via the Graphics Processing Unit (GPU), without the need for plugins. WebGL is standardized and available in all modern browsers.

WebGL is integrated completely into all the web standards of the browser allowing GPU accelerated usage of physics and image processing and effects as part of the web page canvas. WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background.

1.2 How to enable WebGL

To enable WebGL, set the plots `webgl` property to `True`:

```
p = Plot(webgl=True) # for the glyph API
p = figure(webgl=True) # for the plotting API
```

Alternatively, WebGL can be explicitly enabled or disabled on a page by adding `?webgl=1` or `#webgl=0` to the URL.

1.3 Support

Only a subset of Bokeh's objects are capable of rendering in WebGL. Currently this is limited to circle and square marker glyphs. We plan to extend the support to more markers, lines, and other objects such as maps. You can safely combine multiple glyphs in a plot, of which some are rendered in WebGL, and some are not.

The performance improvements when using WebGL varies per situation. Due to overhead in some places of BokehJS, we can currently

not benefit from the full speed that you might expect from WebGL. This is also something we plan to improve over time.

1.4 Notes

- Glyphs drawn using WebGL are drawn on top of glyphs that are not drawn in WebGL.
- When the scale is non-linear (e.g. log), the system falls back to 2D rendering.
- Making a selections of markers on Internet Explorer will reduce the size of the markers to 1 pixel (looks like a bug in IE).

1.5 Example

```
import numpy as np

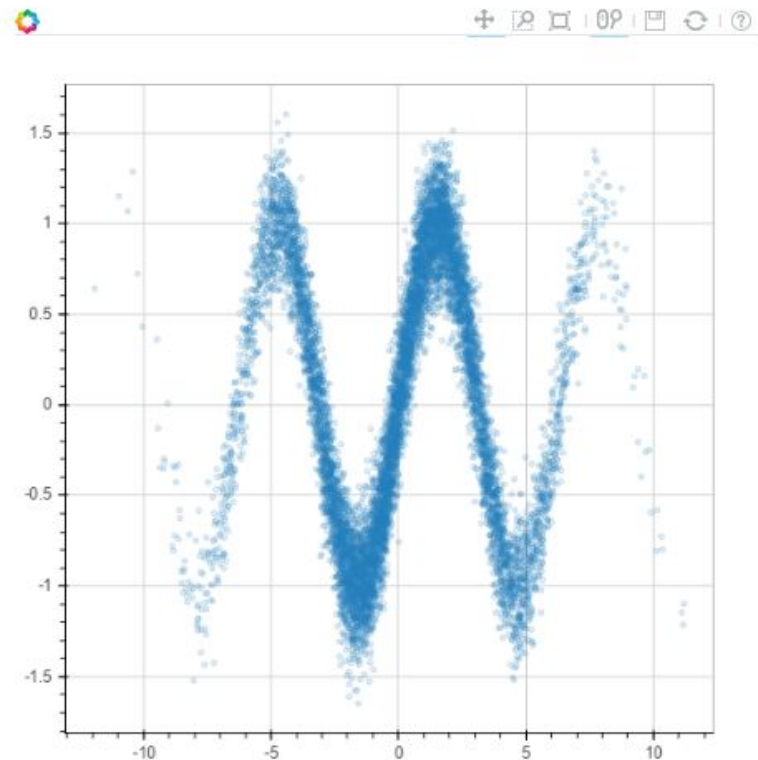
from bokeh.plotting import figure, show, output_file

N = 10000

x = np.random.normal(0, np.pi, N)
y = np.sin(x) + np.random.normal(0, 0.2, N)

output_file("scatter10k.html", title="scatter 10k points (no WebGL)

p = figure(webgl=False)
p.scatter(x,y, alpha=0.1)
show(p)
```



```
import numpy as np

from bokeh.plotting import figure, show, output_file

N = 10000

x = np.random.normal(0, np.pi, N)
y = np.sin(x) + np.random.normal(0, 0.2, N)

output_file("scatter10k.html", title="scatter 10k points (with Web

p = figure(webgl=True)
p.scatter(x,y, alpha=0.1)
show(p)
```

